Mapping performer parameters to synthesis engines

ANDY HUNT† and MARCELO M. WANDERLEY‡

†Department of Electronics, The University of York, Heslington, York YO10 5DD, UK
E-mail: adh@ohm.york.ac.uk
‡Faculty of Music, McGill University, 555 Sherbrooke Street West, H3A 1E3 Montreal, Canada
E-mail: mwanderley@acm.org

This paper considers the issues involved in the design of electronic and computer interfaces, specifically *mapping* – the designed link between an instrument's playing interface and its sound source. It defines the problem area, reviews the literature, and gives examples of specific system mappings. A general model is presented, with the aim of providing a framework for future discussions on what makes an effective mapping. Several guidelines for mapping strategies are given, based on existing work.

1. INTRODUCTION

1.1. Acoustic and electronic instruments

For thousands of years people have made music using acoustic instruments. The performance techniques that have evolved over this time all amount to the (real-time) control of systems that are bound together by physical laws. In most acoustic instruments the sound generation apparatus is intimately coupled to the playing interface. A violin string is both the means of generating vibrations and of constraining them to a particular pitch. The surface of a drum is at the same time the oscillating sound source and the area from which to activate the sound. It is worth noting that in all such cases the musician's energy is physically transformed into the sound energy that is heard by an audience. Also, the instrument's timbre is essentially constrained by its physical construction, but the performer can often make subtle timbral changes in real time.

This may seem like common sense, but it is useful to state at the outset the situation that has existed since the earliest times. Over the years technology has progressed, and mechanisms have been introduced to enhance playing techniques and make possible new forms of sound generation and their real-time control.

There are certain traditional instruments where there is a degree of separation between the source of the oscillations and the human gesture that drives them. For instance, a piano keyboard mechanism consists of keys which are really only part of the playing interface. They do not directly contribute to the sound, as that is made elsewhere by the hammer and strings. However it is still the user's playing energy which is transmitted from the keys to the strings. Because of this separation the piano's timbre is much more tightly constrained than that of, say, a saxophone. However, the piano excels in polyphony and accessibility, and hence remains a universally acknowledged instrument despite its lack of real-time timbral flexibility.

The church organ is a notable exception in a variety of ways, which are worth highlighting when considering the case for designing new instruments. Firstly, it is not the user's energy that creates the sound. Instead the keys and the pedals (the playing interface) are coupled and transmitted to valves that operate the opening and closing of pipes. Yet the energy to sound those pipes comes from bellows, traditionally driven by another person, yet mostly now operated electrically. In large churches and modern organs the electricity is also used to control pipes that are further away, or in more complex combinations than mechanical coupling would allow. The use of an external energy supply, in combination with the complex routing of keys to pipes, results in an instrument that can produce enough sound energy to fill a large space with rich timbres, controllable by the player.

The organ shares a series of attributes with most modern electronic and computer instruments. It has an interface which is completely isolated from the sound source that it is driving. The instrument designer defines how the playing interface connects to the sound source. The player can use specific controls to change the timbre (on the organ this is done by the use of the stops, the swell pedal and coupling stops). The energy to make the sound comes from an external supply, not the player's energy.

Those of us who design electronic instruments, and are sometimes faced with unfavourable comparisons to acoustic instruments, should perhaps take comfort in this particular comparison. If such a widespread and refined instrument as the church organ is popular, despite the fact that it deviates wildly from the control methods of most other acoustic instruments, then there must be a place *per se* for novel electronic instruments in society. Among the reasons for the creation of new instruments are the real-time control of new sound-worlds, and the control of existing timbres through alternative interfaces to enable individuals in the spontaneous creation of music whatever their physical condition (Abbotson,

DOI:10.1017/S1355771802002030

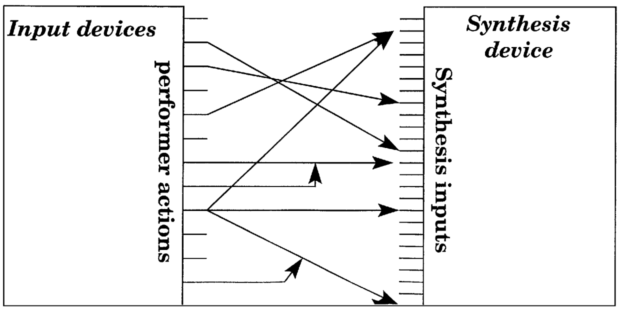

Figure 1. A representation of the mapping layer (arrows) between performance data and the input variables of a synthesis engine.

Abbotson, Kirk, Hunt and Cleaton 1994, Hunt, Kirk, Abbotson and Abbotson 2000).

Moreover, in the case of new electronic instruments, there are no restrictions about the oscillation mechanism. Because sound is generated by an electronic device (synthesizer or computer), the physics of vibrating structures that were determinant to the forms and features of acoustic instruments do not necessarily play any role in new instruments. This brings up a whole new set of opportunities in the design, since physical constraints are almost non-existent.

However, how are we to go about the act of designing a new instrument? Are there guidelines available? Are there things that can be learnt about a human being's inherent interaction with real-time acoustic systems that will make it easier to design and build better electronic instruments? This paper considers the role of the instrument's designer, who has to specifically define the relationship between the playing interface and the sound source.

1.2. Mapping

The term '*mapping*' is used widely to indicate the mathematical[1] process of relating the elements of one data set onto another. In computer music, mapping is often used in relation to algorithmic composition, where a parameter with a particular set of values is scaled or transformed so that it can be used to control another parameter (Winkler 1998). An example of this might be the 'mapping' of the output of a random-number generator onto a probability table in order to control the choice of pitch in a composition. In this paper we define *mapping* as the act of taking real-time performance data from an input device[2] and using it to control the parameters of a synthesis engine.

Nowadays there are many physical controllers available, and many hardware and software synthesis environments. There is a temptation for engineers to directly connect each element of the interface to a synthesis parameter in one-to-one relationships (e.g. slider position = volume control). This is a mapping, whether it has been thought about much or not. The purpose of this paper is to raise the profile of this mapping section in electronic instruments, to see what we can learn from what has already been done, and to attempt to formulate some basic guidelines for instrument designers.

2. MAPPING LITERATURE REVIEW

There have already been attempts to address the role of mapping in computer music performance. An early example is discussed by Abbot in 1982 (Pennycook 1985), which refers to the role of parameter mapping in the performance of computer music.

2.1. Ways of implementing mapping

The main question to be solved is related to the actual choice of which mapping strategy to implement. Considering mapping as part of an instrument, one can deduce two main directions from the analysis of the existing literature:

- The use of generative mechanisms, such as neural networks to perform mapping.
- The use of explicitly defined mapping strategies.

The main difference between these two directions lies in the chosen approach to mapping. The former uses a method that provides a mapping strategy by means of internal adaptations of the system through training or the selection of the most important features among the set of signals. The latter proposes mapping strategies which explicitly define the input-to-output relationships.

Mapping using neural networks allows the designer to benefit from the self-organising capabilities of the model. Compared to the use of neural networks, explicitly defined mapping strategies present the advantage of keeping the designer in control of the design of each of the instrument's component parts, therefore providing an understanding of the effectiveness of mapping choices in each context.

[1] Strictly speaking, in mathematics, mapping is defined as the assignment of every element of a set to an element of the same set or another set.

[2] In computer music, the words input device, gestural controller, control surface and hardware interface are used interchangeably.

2.1.1. Mapping using neural networks

The use of neural networks allows for designers to implement mappings through the adaptation of the network to the specific performance context. Lee and Wessel (1991) proposed the use of multi-layer neural networks trained by back propagation to control sound synthesis. One application consisted of the use of a neural network as a multidimensional map to dynamically control a timbre using data from a MIDI keyboard. Fels and Hinton (1995) presented an interface to map gestures to speech synthesis, using a gesture-to-formant model, called Glove-TalkII. In this system, three feed-forward neural networks were used to map parameters available from a Cyberglove, a Polhemus sensor and a Contact glove, onto ten continuous parameters of a speech signal. Finally, Modler (2000) used neural networks to map hand gestures through a sensor glove to musical parameters in an interactive music performance and virtual reality environment.

2.1.2. Explicit mappings: general mappings between two sets of parameters

The available literature generally considers mapping of performer actions to sound synthesis parameters as a *few-to-many relationship* (Lee and Wessel 1991). This is mostly true in the case of synthesis by *signal models*, such as source-filter or additive synthesis, where sometimes hundreds of variables represent the available synthesis inputs. For example, the user control of additive synthesis consists of mapping a limited number of human control parameters onto the amplitude and frequency of several, perhaps hundreds of oscillators.

Considering two general sets of parameters, three intuitive strategies relating the parameters of one set to the other can be devised as:

- *one-to-one*, where one synthesis parameter is driven by one performance parameter,
- *one-to-many*, where one performance parameter may influence several synthesis parameters at the same time, and
- *many-to-one*, where one synthesis parameter is driven by two or more performance parameters.

In addition, a possible combination of the above basic strategies could be termed *many-to-many*. The classification of mapping into these three basic types provides a general method of thinking about how two sets of parameters relate to each other. It focuses on the essence of how the elements of one set are translated into the other. Similar classifications have been presented in the musical literature in different ways by Ryan (1990), Garnett and Goudeseune (1999), and Rovan, Wanderley, Dubnov and Depalle (1997), who have identified the three basic categories using the words *one-to-one*, *convergent* (many-to-one) and *divergent* (one-to-many).

Concerning explicit mappings between two sets of parameters, Bowler, Purvis, Manning and Bailey (1990) presented a general way to map performance parameters to synthesizer control parameters through the proposition of an interpolation scheme that assured continuity between two sets of control parameters. Choi, Bargar and Goudeseune (1995) proposed a mapping of a point in a control space to a point in a phase space by identifying the co-ordinates of the cell it is in, and relating them to the corresponding cell and co-ordinates in the phase space. Garnett and Goudeseune (1999) referred to a refinement of this through the use of a general method from geometric topology, called simplicial interpolation. It is interesting to notice that the authors suggest the extension of such geometric mapping to *perceptual* parameters (instead of from control parameters to synthesis ones). A perceptual parameter is basically an abstraction of the set of raw control inputs (e.g. modulation amplitude) into a concept that can be perceived as sound (e.g. ‘brightness’). This idea is basically similar to the one presented by Metois (1996).

Wanderley, Schnell and Rovan (1998) proposed an extension to this model by using *abstract* instead of *perceptual* parameters. In Wanderley *et al.* (1998) the mapping of control parameters to synthesis inputs is divided into two layers: control parameters to abstract parameters, and abstract parameters to synthesis inputs. This will be explained in detail in section 4.

Mulder, Fels and Mase (1997) suggested the use of geometrical shapes as an attempt to reduce the cognitive load of manipulating several simultaneous parameters. Their approach consisted of focusing on continuous parameter changes represented by gestures produced by the user. In other words the real-time complex movements of the user are mapped onto geometrical forms whose features represent sound control parameters. This allowed users to think about common manipulation metaphors (e.g. claying, sculpting, etc.) whilst controlling the evolution of previously generated sound – a *score-level control metaphor* (Orio, Schnell and Wanderley 2001).

3. INSTRUMENTAL MAPPING EXAMPLES

To aid us in our quest for some general mapping guidelines, we will now review pieces of work in which the authors have sought to specifically address the issue of mapping. In other words, although there are clearly many computer-based performance pieces and instruments in existence, we shall now focus on those works whose main purpose was to explore the role of mapping in real-time instrumental control. The first of these concerns the design of a new instrument by modelling the mapping techniques of a corresponding acoustic instrument. The second explores the fundamental issues of how humans interact when faced with many parameters to control in real time. The third takes as its premise that many musicians wish to build upon the many years of control they have invested in their acoustic instruments by using them to control new sound worlds and visual media.

3.1. Inspiration from acoustic instruments

Rovan *et al.* (1997) presented an implementation of a performance system using a MIDI wind controller and the morphing of models of acoustic wind instruments in additive synthesis. The idea was to challenge the main directions in digital musical instrument design, i.e. 'input device design' and 'research into new synthesis algorithms', and focus on the importance of different mappings using off-the-shelf controllers and standard synthesis algorithms.

Different mapping strategies were applied to the simulation of traditional acoustic single reed instruments using the controller, based on the actual functioning of the single reed. The three basic strategies presented in section 2.1.2 – *one-to-one*, *divergent*, and *convergent* – were used in order to propose different mappings, from a simple one-to-one to complex mappings simulating the physical behaviour of the clarinet's reed.

Figure 2 shows the three experiments with mapping presented in Rovan *et al.* (1997). From top to bottom, the modelling of the digital musical instrument moves from a simple recorder-like instrument to a more complex single-reed-like instrument. In fact, in the first model, both the loudness and dynamics (spectrum content) are determined by simple one-to-one relationships – they are actually directly proportional to the independent controlling variables. In the second layer, loudness is a function of *both* lip pressure *and* airflow, both coupled in a nonlinear relationship. This follows the physics of single reed instruments, where the actual quantity of air to pass through the reed is dependent on both variables. Finally, in the third mapping scheme, both loudness and dynamics respond to functions of two variables in a similar way to what happens in single-reed instruments.

From experiments with the original system and its subsequent implementations (Wanderley 2001), the authors showed that the use of different mappings *did* influence the expressivity obtained during the playing, without modifications to either the input device or the synthesis algorithm.

3.2. Complex mapping for arbitrary interfaces

The first author carried out an investigation into the psychology and practicality of various interfaces for real-time musical performance (Hunt 1999). The main part of this study took the form of a major series of experiments to determine the effect that interface configuration had on the quality and accuracy of a human player's performance. The full thesis is available for download online (Wanderley 2000), and the details of the theory, experiments and results have been published (Hunt and Kirk 2000). They are summarised here, and discussed further within the context of this paper – i.e. aiming to find general principles for mapping strategies in musical performance.

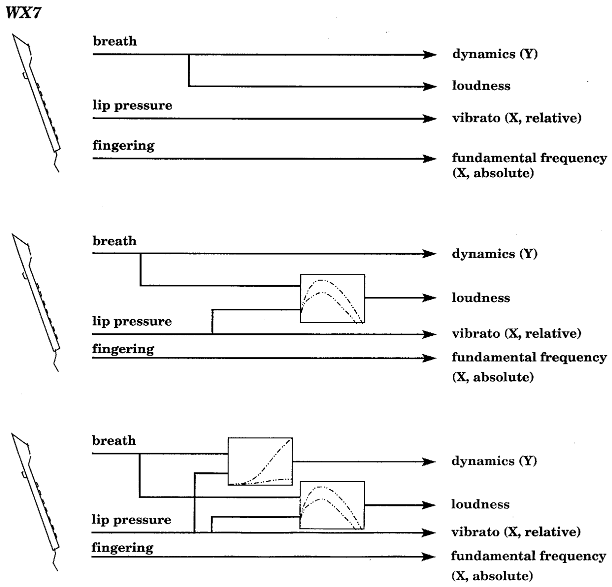

Figure 2. Mappings implemented in Rovan *et al.* (1997). The control variables from the WX7 were mapped to a bi-dimensional timbre space where the X-axis represented the pitch and the Y-axis the spectrum distribution information of additive models of clarinet samples.

3.2.1. Experimental outline

Essentially the experiment was devised to compare several humans' performances (of a range of relatively simple musical tasks) on different interfaces. The progress of the participants *over time* was also studied, and this has subsequently shed light on the learning process and how it relates to the mapping employed in the interfaces.

The musical tasks took the form of a 'listen and copy' test. The computer played a sound, where any combination of four sonic parameters was altered, then the user was asked to recreate the sound as accurately as possible on the interface in front of them. The four sonic parameters chosen were continuous pitch, amplitude, harmonic content (controlled by a simple low-pass filter) and stereo panning position. These four data streams formed a common metric for every test (sixteen participants, over several sessions on three interfaces, each tested by twenty-four tasks). The tasks increased in complexity from step-changes in one parameter only (such as two consecutive pitches), through to continuous changes in all four parameters containing musical expression.

3.2.2. Interfaces under test

Three interfaces were chosen for the experiment. The first (known as the *mouse* interface) represented a commonly used paradigm – the user manipulating on-screen sliders with a computer mouse. Each slider controls one of the four parameters, and thus in common with many engineering solutions there was a one-to-one mapping between each on-screen control area and the parameter being controlled.

The second interface (*sliders*) consisted of a set of four physical sliders (rather like a small mixing desk) linked, via MIDI, to the on-screen sliders. Again, each of these sliders was mapped in a one-to-one fashion onto the four sonic parameters. The main difference between this interface and the first was its physicality. Users had the implicit option of being able to operate more than one slider at the same time. They also had no real need to look at the screen, though most chose to do so constantly.

The third interface (*multiparametric*) used more complex mapping techniques, yet used the same physical input devices (MIDI sliders and a mouse) as the control devices. One hand is required to control the mouse, whilst the other operates two of the physical sliders.

3.2.3. Guidelines for multiparametric mapping

The mappings were devised to be *non-obvious* to the first-time user; a situation encountered by everyone learning a new musical instrument. However, they were by no means completely arbitrary, as they were based on the following observations of most acoustic instrument interfaces:

- Energy is required for amplitude.
- Two hands are used.
- Complex mappings are present – changes of one parameter inflect the others.
- Timbre is controlled in a non-direct manner.

Discussing each of the above points in turn:

(1) *Energy is required for amplitude*. In most acoustic instruments the human operator has to inject energy or 'excite' the system before it will operate, and must continue to supply energy to keep it going. This action normally takes place by bowing, plucking, blowing or striking. The output amplitude of the system is mostly in direct proportion to the energy of the input gesture. Then, once the energy has been injected it is steered through the system or damped (dissipated) in order to make sound.

(2) *Two hands are used*. In most instruments the above two energy-based operations (inject/excite and steering/damping) are often carried out by different

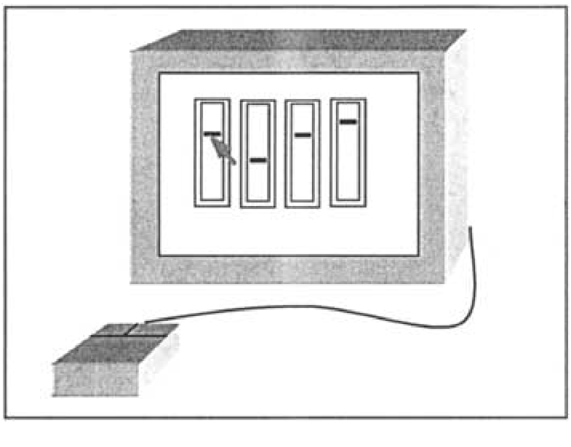

Figure 3. The Mouse & Sliders Interface.

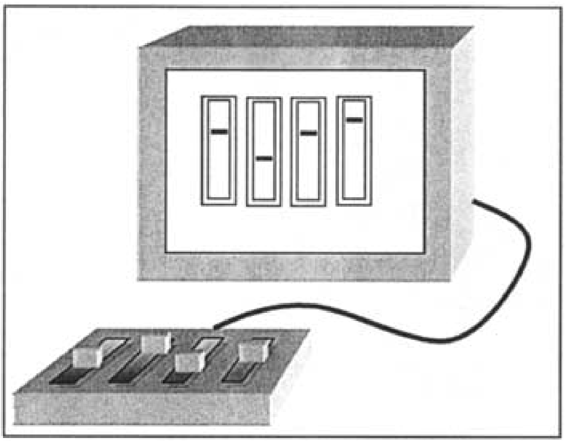

Figure 4. The Physical Sliders Interface.

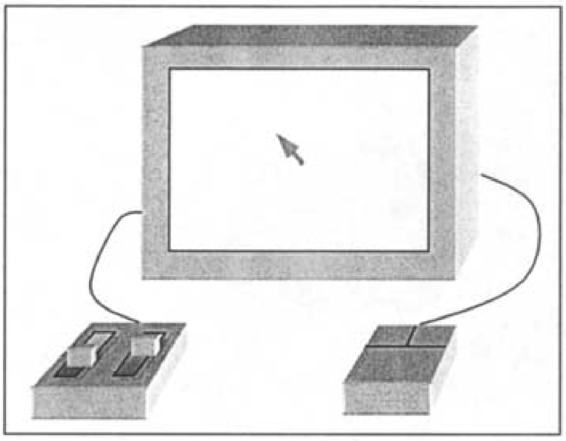

Figure 5. The Multiparametric Interface.

'limbs' (e.g. bowing with one arm and fingering notes with another, or blowing with the lungs whilst squeezing with the lips and fingering the note patterns). As stated at the outset of this paper, the church organ differs in that its energy is produced by an external source, but the keyboard and pedals provide gating of this energy, and the swell pedal provides a continuously controllable form of damping.

(3) *Complex mappings are present – changes of one parameter inflect the others*. Let us consider the correspondences between each input control and the controllable parameters in acoustic instruments. For example, picture a violin and ask, 'where is the volume control?' There is no single control; rather a combination of inputs such as bow-speed, bow pressure, choice of string and even finger position. This is an example of a 'many-to-one' mapping, where several inputs are needed to control one parameter. Again, considering the violin, ask, 'which parameter does the bow control?' It actually influences many aspects of the sound such as volume, timbre, articulation and (to some extent) pitch. This is therefore an example of a 'one-to-many' mapping. Human operators expect to encounter complex mappings, and yet so often engineers provide nothing but 'one-to-one' correspondences.

(4) *Timbre is controlled in a non-direct manner*. Unlike a church organ – where the user can select the timbre by a series of stops with a *structural modification* gesture (Cadoz and Wanderley 2000) – most acoustic instruments do not have a specific control for timbre. Instead the physical nature of the system ensures that it is varied by almost every control parameter. This is a special case of point (3) (above) where many input parameters combine to affect a particular feature of the system. Players of acoustic instruments are used to this way of working, and so it has become an unspoken part of the musician's psyche over the centuries.

And so to the third interface (multiparametric) and the mapping strategies that were implemented to adhere to the observations outlined above. There is nothing for the user to look at on the screen – it is just an area within which to move the mouse pointer. Sound is only made when the mouse is moved. The sound's volume is proportional to the speed of mouse movement. This ensures that the user's physical energy is needed for any sound to be made, and that the amount of energy has an effect on the amplitude of the sound. In addition, the volume, pitch, timbre and panning are controlled by combinations of the mouse position and the position of two sliders, as shown here:

> Volume = speed of mouse + mouse button pressed + average position of two sliders.
> Pitch = vertical position of the mouse + speed of movement of slider no. 2.
> Timbre = horizontal position of the mouse + difference in the two slider positions.
> Panning = position of slider no. 1.

This implements several many-to-one mappings. Simultaneously there are various one-to-many mappings (e.g. slider 1 affects volume, timbre and panning). Two limbs are used, as the player has to use two hands – one on the mouse, one on the sliders.

3.2.4. Quantitative results

Every test result was stored on the computer and later given a score by a human marker and moderator, giving a percentage accuracy accumulated over 4,000 tests over a period of several weeks. It became clear that the multiparametric interface allowed users to perform in a different manner to the other two interfaces:

- The test scores were much higher than the other two interfaces, for all but the simplest tests.
- There was a good improvement over time across all test complexities.
- The scores got better for more complex tests!

This last result may seem rather counter-intuitive at first sight; that people performed better on the harder tasks. However, this brings into question the definition of a 'hard task'. If an interface allows the simultaneous control of many parameters, maybe it really is easier to perform the more complex tasks, and harder to accurately isolate individual parameters.

The mouse interface, on the other hand, allowed the direct control of an individual parameter, but was almost impossible to control for anything more complex. The sliders interface, whilst it physically allowed people to control multiple parameters, forced the user to mentally 'strip apart' the control task into separate control streams. This caused a form of cognitive overload which users generally found restricting and frustrating.

3.2.5. Qualitative discussion

Qualitative analysis was also carried out by interviewing every subject after each set of tests on each interface. They were asked how they felt about their performance and the interface. Towards the end of the session they were asked to sum up how they had done overall and to compare the different interfaces.

The interviews indicated that the majority of users enjoyed playing the multiparametric interface and thought (quite enthusiastically) that it had the best long-term potential. The following three points are indicative of the most commonly reported comments summarised from the users. They indicate that there was something about the multiparametric interface that was entertaining and engaging, perhaps because it allowed a certain amount of spatial thinking.

- The multiparametric interface allowed people to think gesturally, or to mentally rehearse sounds as shapes.
- The majority of users felt that the multiparametric interface had the most long-term potential. Several people commented that they would quite like to continue to use it outside the context of the tests!
- Several users reported that the multiparametric interface was fun.
- In contrast the sliders interface often elicited the opposite response and the majority of people found it confusing, frustrating or at odds with their way of thinking. This was often focused on the requirement to mentally break down the sound into separate parameters. Maybe the test subjects were experiencing cognitive overload with the harder tasks on this particular interface.

Since both the sliders and multiparametric interfaces allowed the user to have continuous control over all four sound parameters, we can conclude that the above differences can be accounted for by the specific parameter mappings alone. In other words:

> *Mapping strategies which are not one-to-one, and which utilise a measure of the user's energy under the control of more than one limb (or body part), can be more engaging to users than one-to-one mappings.*

3.3. Mapping to a graphical synthesizer

In today's multimedia world, artists are increasingly encountering the problem of how to map creatively from one media to another – particularly how to define the relationship between sound and image. While composers of music to accompany film have been wrestling with this issue for nearly a century, they have always essentially brought a sonic composition into temporal proximity to a pre-composed visual composition. Each medium is (usually) composed by different people, and recorded in different times and places. The widespread availability of fast digital computing allows the real-time creation of both media. Some of the fundamental issues involved in the creative design of this process have been published (Hunt, Kirk, Orton and Merrison 1998), but they all amount to the composer being in charge of the mapping between the performer, the score and the output media.

The RIMM project (Real-time Interactive MultiMedia) (RIMM 2001) involved a single saxophonist playing live on stage, supported by a computer network and hardware infrastructure which processed his sound, synchronised with a score, and produced graphical images projected onto a textured screen. The composer conceived the sound score and a graphical designer worked with him to produce a corresponding graphical 'storyboard'. The project involved a series of technical challenges for the team to create an almost invisible set of sensors with supporting hardware and software, to enable the lone player to work unhindered on stage. However, once again, it seemed that the main artistry was to be found in the coupling of the composers' ideas with the mapping strategies that linked the different parts of the system together.

In the RIMM project the first author was involved in defining the mapping strategies between the performer, the computer score and the graphics. In the early stages of the project it was necessary to define how the graphics would:

(1) be affected by the performer in real time,
(2) be controlled by the computer score, and
(3) operate independently of both (using a graphical rule-base).

The saxophonist had built up almost two decades of performance subtlety on his instrument, and so it was felt that much of the complex acoustic mapping was inherently present in his existing performer–instrument system. The main task was therefore to convert the instrument's sound, and the player's performance gestures, into new sounds and graphical control. Note how different the situation would be if the player were using a novel electronic sax-like controller, and the entire sound synthesis system had to be defined and mapped.

The team devised a method of communicating ideas, and configuring the system, based on the use of meaningful or perceptual parameters (Oppenheim, Anderson and Kirk 1993). For example, it was established that there should be a perceivable relationship between the saxophone's 'lift' (the height of the bell off the floor), the 'brightness' of the computer sound, and the graphical 'colour temperature'. In practice, this involved a good deal of mathematical scaling and cross-coupling, but it was very helpful to hold onto the concept of these tangible parameters. This led to the idea that we were actually working on several simultaneous mapping layers. Each stage of figure 6 is now considered in turn.

Data is first gathered from the sensors. Sometimes this comes from an electrical sensor (e.g. an accelerometer produces voltages representing acceleration in the x-plane and y-plane) via a digital-to-analogue converter (DAC). At other times the data comes from the processed sound of the instrument (e.g. a rolling average amplitude level). Data from all the sensors are converted into a 'normalised' set of single values. For example, the motion of a mouse would be split into two separate streams (for x and y position) each scaled as a number between 0.0 and 1.0. Switches are defined as on/off (1.0/0.0).

The first true stage of mapping takes these normalised data streams and extracts from them meaningful or tangible parameters. This could be as simple as 'mouse x position' (which involves little processing) or more complex concepts such as 'player's energy' (which might be a blend of notes played per second, the speed of movement of the instrument, and the average amplitude level).

The second stage of mapping allows the connection of 'user-side' perceptual parameters to 'system-side' ones, e.g. the connection of 'sax height' to 'brightness' (where the height is clearly gathered from the performer, but brightness is a descriptive input to a synthesis process). Note that, at this stage, several sub-systems can be operational; in our case a synthesis unit, a sound spatialisation engine, and a graphics generator.

The final stage in the mapping process consists of transforming the 'system-side perceptual parameters' into the available inputs to the available sub-systems. For example, 'brightness' would need to undergo an interesting and non-trivial mapping process to control the perceived brightness of a frequency modulation (FM) instrument.

The above mapping scheme allowed us to specify the compositional effect at a *meaningful* level, and to make sensible and consistent mappings between different media sub-systems, such as an audio synthesizer and a graphics engine. It is interesting here to mention other developments based on real-time control of sound synthesis and transformation by single wind instrument performers. Specifically for the case of the clarinet, these include pieces by Pennycook (1991) and Rovan (1997), as well as applied research by Egozy (1995). These differ in the gestural approach followed. Pennycook used special sensors adapted to the instrument which were used at specific moments in the piece, whereas Rovan used an infra-red proximity sensor to measure deliberate movements of the performer. Therefore, considering the two pieces, deliberate gestures – instrumental in Pennycook's piece and ancillary in Rovan's one – were used as the main control variables (Goldstein 1998, Cadoz and Wanderley 2000). On the other hand, Egozy proposed a system to analyse the instrument's sound to derive control parameters in real time without the demand for extra (non-traditional) performance gestures. But one can also consider the acquisition of non-deliberate performer gestures, which poses yet another set of questions: Are these gestures consistent across performances of the same instrumentalist? Are there similar gestural patterns across different performers?

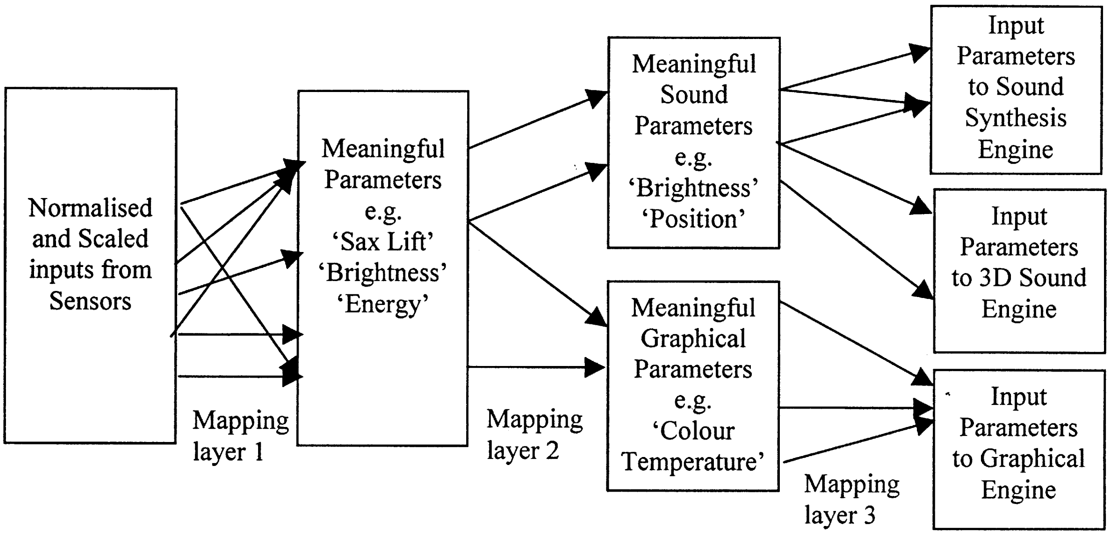
Normalised and Scaled inputs from Sensors
Mapping layer 1
Meaningful Parameters e.g. 'Sax Lift' 'Brightness' 'Energy'
Mapping layer 2
Meaningful Sound Parameters e.g. 'Brightness' 'Position'
Meaningful Graphical Parameters e.g. 'Colour Temperature'
Mapping layer 3
Input Parameters to Sound Synthesis Engine
Input Parameters to 3D Sound Engine
Input Parameters to Graphical Engine

Figure 6. Mapping layers within the RIMM project.

Are there quantitative measures of these movements? (Wanderley 2001).

4. A MODEL FOR INSTRUMENTAL MAPPING

In Wanderley and Depalle (1999) and Hunt, Wanderley and Kirk (2000) we proposed a two-layer mapping strategy for general use based on Wanderley *et al.* (1998). This model builds on previous developments (Wessel 1979, Metois 1996, Mulder *et al.* 1997) by considering the *explicit role of the chosen mapping strategies* in each layer, as well as the choice of intermediate parameters.

In this model, the first layer is *interface-specific*, since it converts the incoming sensor information into a set of chosen (intermediate or abstract) parameters. These parameters can either be perceptually relevant, as in Metois (1996) or Garnett and Goudeseune (1999), or derived from other interaction metaphors (Mulder *et al.* 1997). These are then mapped – in a second independent mapping layer – onto the specific controls needed for a particular synthesis engine.

Compared to the models proposed in Bowler *et al.* (1990) or in Choi *et al.* (1995), where input parameters are directly mapped into synthesis parameters, the advantages of this two-layer model are as follows. A change in *either* the input device *or* the synthesis engine would require the modification of *either* the first *or* the second mapping layers, provided its output parameters can be mapped to the originally defined abstract parameters. This means that the first mapping layer in the model is a function of the given input device and the chosen abstract parameters. The same can be considered for the second layer: given the same input device and abstract parameters (fixed first mapping layer), the synthesis engine can be changed by changing the second mapping layer, e.g. from additive synthesis (Wanderley *et al.* 1998) to FM synthesis (e.g. Vertegaal and Bonis 1994).

The use of explicit models for designing mappings is not only an important task for instrument designers. In fact, it should be noted that even if composers state that they are not using mapping techniques explicitly, they would still have to define all of the following stages if the piece involves live electronic instruments. We claim here that it is easier, more powerful and more artistically productive to consider these stages as separate entities, some of which can be offered out to technicians to complete. The proposed mapping layers are:

(1) Extraction of meaningful performance parameters (an optional extra layer needed in certain circumstances, e.g. video input, but also for deriving performance-relevant parameters, such as the player's *energy* input to the system).
(2) Connection of performer's (meaningful) parameters to some intermediate representation set of parameters (for instance, perceptual or abstract).
(3) Decoding of intermediate parameters into system-specific controls.

This model also has the added advantage that the integrated control of several media can be accommodated. In this case there would be several connections between layers (2) and (3), allowing abstract parameters to be linked for each of the sub-systems (e.g. synthesis, sound spatialisation and graphics).

5. DISCUSSION

The question then arises about what specific mappings to use in each of these sections of the model presented above. The works detailed previously in this paper imply that complex mappings are more effective at eliciting a good performance from a human player when the performer is confronted with multiparametric tasks, more so than a series of one-to-one mappings which is often

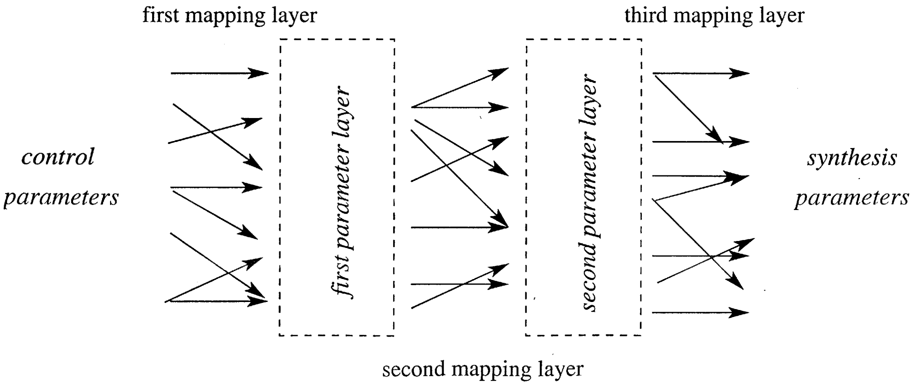

Figure 7. Three-layer general mapping model.

the engineering default. The advantage of the above model is that a designer would have to work very hard indeed to ensure a set of one-to-one mappings from input sensor to output controls!

In section 3.2 we have outlined a set of suggested guidelines involving the use of the player's energy, the use of different body parts to inject and transform the energy, and the 'hiding' of certain key parameters (such as the instrument's timbre, volume and pitch) as a combination of inputs. The use of models consisting of multiple mapping layers – established by the definition of intermediate parameters – allows for the accommodation of different media in the same digital musical instrument design. Other questions to be discussed when approaching the definition of mapping strategies for real-time computer music include:

- *The relationship of mapping complexity to task complexity and to user expertise*, i.e. amount of training.
- *The match of input device and mapping to task complexity*. Jacob *et al.* (1994) discuss the match of the input device to the task to be executed, showing that tasks perceived as integral are better performed with the use of devices allowing navigation between several variables. Hunt and Kirk (2000) showed that in tasks demanding holistic thinking, complex mappings outperform simple ones, whilst simple mappings are better suited to analytical thinking. General studies need to be performed to confirm these findings in different circumstances.
- *The definition of musical tasks* (Orio *et al.* 2001). How to isolate performance variables in a musical task in order to run commonly accepted statistical tests and still keep the musical characteristics of the task, i.e. its multiparametric nature.
- *The importance of mapping consistency over time.* It has been claimed (Wessel 1991) that instruments that present the ability to adapt to the user's playing style can eventually become more expressive. The potential problem with adaptive instruments is the reduction of demands on the amount of effort the user needs to spend in order to master the instrument – and it has been claimed that effort and expression are somewhat related (Ryan 1992). In other words, to what extent are *dynamic mappings* of interest to instrumental designers and to skilled performers of digital musical instruments? If in the context of interactive composition this question could perhaps be easier to answer due to the inherent temporal evolution of the musical material, in the instrumental case it may perhaps become counterproductive if it demands less effort from a performer. It could also be considered that a system which reconfigures itself is a case of 'constantly moving the goalposts'. In other words the human mind–body system is extremely good at adapting to fixed physical challenges, and learning their subtlety. How will this be affected if the physical challenge is constantly modifying itself?

Therefore, there is a need for a deeper discussion of mapping design and its effectiveness in the various contexts related to computer music. An effort in this direction is currently being developed in the Interactive Systems and Instrument Design in Music Working Group, supported by the International Computer Music Association (ICMA) and the Electronic Music Foundation (EMF). Since 2000 we have been collecting references about the various aspects of mapping and there is already a wealth of information available on the group's website (Hunt and Wanderley 2000), including introductory material with reading guidelines, a complete list of references and links to resources available online.[3]

6. CONCLUSIONS

The art of mapping is as old as acoustic instrument design itself, but it is only since the invention of real-time electronic instruments that designers have had to explicitly build it into each instrument. This paper shows that there is much to be done in order

[3] We welcome suggestions about papers and works related to mapping.

to understand this subject fully, yet a good deal of work has been carried out in recent years. Much of this work shows that the subject of mapping must be addressed in real-time systems, if the next generation of electronic and computer instruments are going to allow a similar level of sonic and performance subtlety to their acoustic ancestors.

REFERENCES

Abbotson, M., Abbotson, R., Kirk, R., Hunt, A., and Cleaton, A. 1994. Computer music in the service of music therapy: the MIDIGRID and MIDICREATOR systems. *Medical Engineering Physics* **16**(May): 253.

Bowler, I., Purvis, A., Manning, P., and Bailey, N. 1990. On mapping N articulation onto M synthesiser- control parameters. *Proc. of the 1990 Int. Computer Music Conf.*, pp. 181–4. San Francisco, CA: International Computer Music Association.

Cadoz, C., and Wanderley, M. M. 2000. Gesture-music. In M. Wanderley and M. Battier (eds.) *Trends in Gestural Control of Music*. Ircam – Centre Pompidou.

Choi, I., Bargar, R., and Goudeseune, C. 1995. A manifold interface for a high dimensional control space. *Proc. of the 1995 Int. Computer Music Conf.*, pp. 385–92. San Francisco, CA: International Computer Music Association.

Egozy, E. 1995. Deriving Musical Control Features from a Real-time Timbre Analysis of the Clarinet. Master's thesis, Massachusetts Institute of Technology.

Fels, S., and Hinton, G. 1995. Glove Talk II: an adaptative gesture-to-formant interface. *Proc. Conf. on Human Factors in Computing Systems (CHI'95)*, pp. 456–63.

Garnett, G., and Goudeseune, C. 1999. Performance factors in control of high-dimensional spaces. *Proc. of the 1999 Int. Computer Music Conf.*, pp. 268–71. San Francisco, CA: International Computer Music Association.

Goldstein, M. 1998. Gestural coherence and musical interaction design. *Proc. of the 1998 IEEE Int. Conf. on Systems, Man and Cybernetics (SMC'98)*, pp. 1,076–9.

Hunt, A. 1999. *Radical User Interfaces for Real-time Musical Control*. D.Phil. thesis, University of York, UK.

Hunt, A., and Kirk, R. 2000. Mapping strategies for musical performance. In M. Wanderley and M. Battier (eds.) *Trends in Gestural Control of Music*. Ircam – Centre Pompidou.

Hunt, A., Kirk, R., Abbotson, M., and Abbotson, R. 2000. Music therapy and electronic technology. *Proc. EuroMicro Conference 2000*, vol. II, pp. 362–7.

Hunt, A., and Wanderley, M. M. (eds.) 2000. Mapping of control variables to musical variables. *Interactive Systems and Instrument Design in Music Working Group*. http://www.igmusic.org

Hunt, A., Wanderley, M. M., and Kirk, R. 2000. Towards a model for instrumental mapping in expert musical interaction. *Proc. of the 2000 Int. Computer Music Conf.*, pp. 209–11. San Francisco, CA: International Computer Music Association.

Hunt, R., Kirk, R., Orton, R., and Merrison, B. 1998. A generic model for compositional approaches to audiovisual media. *Organised Sound* **3**(3): 199–209.

Jacob, R., Sibert, I., McFarlane, D., and Mullen, M. P. 1994. Integrality and separability of input devices. *ACM Transactions on Human–Computer Interaction* **1**(1): 3–26.

Lee, M., and Wessel, D. 1992. Connectionist models for real-time control of synthesis and compositional algorithms. *Proc. of the 1992 Int. Computer Music Conf.*, pp. 277–80. San Francisco, CA: International Computer Music Association.

Metois, E. 1996. *Musical Sound Information – Musical Gestures and Embedding Systems*. Ph.D. thesis, Massachusetts Institute of Technology.

Modler, P. 2000. Neural networks for mapping gestures to sound synthesis. In M. Wanderley and M. Battier (eds.) *Trends in Gestural Control of Music*. Ircam – Centre Pompidou.

Mulder, A., Fels, S., and Mase, K. 1997. Empty-handed gesture analysis in Max/FTS. In A. Camurri (ed.) *Kansei, The Technology of Emotion. Proc. of the AIMI Int. Workshop*, Genoa: Associazione di Informatica Musicale Italiana, 3–4 October 1997.

Oppenheim, D., Anderson, T., and Kirk, R. 1993. Perceptual parameters – their specification, scoring and control within two software composition systems. In *Proc. of the 1993 Int. Computer Music Conf.* San Francisco, CA: International Computer Music Association.

Orio, N., Schnell, N., and Wanderley, M. M. 2001. Input devices for musical expression: borrowing tools from HCI. Presented at the *Workshop New Interfaces for Musical Expression*, ACMCHI 2001.

Pennycook, B. 1985. Computer-music interfaces: a survey. *Computing Surveys* **17**(2): 267–89.

Pennycook, B. 1991. The PRAESCIO Series: composition driven interactive software. *Computer Music Journal* 15(3).

RIMM. 2001. The Real-time Interactive MultiMedia project. http://www.york.ac.uk/res/rimm/

Rovan. J. B. 1997. Continuities for Clarinet and Electronics.

Rovan, J. B., Wanderley, M. M., Dubnov, S., and Depalle, P. 1997. Instrumental gestural mapping strategies as expressivity determinants in computer music performance. In A. Camurri (ed.) *Kansei, The Technology of Emotion. Proc. of the AIMI Int. Workshop*, pp. 68–73. Genoa: Associazione di Informatica Musicale Italiana.

Ryan, J. 1991. Some remarks on musical instrument design at STEIM. *Contemporary Music Review* **6**(1): 3–17.

Ryan, J. 1992. Effort and expression. *Proc. of the 1992 Int. Computer Music Conf.*, pp. 414–16. San Francisco, International Computer Music Association.

Vertegaal, R., and Bonis, E. 1994. ISEE: an intuitive sound editing environment. *Computer Music Journal* **18**(2): 212–29.

Wanderley, M. M. (ed.) 2000. Interactive Systems and Instrument Design in Music Workgroup. http://www.igmusic.org

Wanderley, M. M. 2001. *Performer–Instrument Interaction: Applications to Gestural Control of Sound Synthesis*. Ph.D. thesis, University Paris VI.

Wanderley, M. M., and Depalle, P. 1999. Contrôle gestuel de la synthèse sonore. In H. Vinet and F. Delalande (eds.) *Interfaces homme–machine et création musicale*, pp. 145–63. Paris: Hermµs Science Publications.

Wanderley, M. M., Schnell, N., and Rovan, J. B. 1998.

Escher – modeling and performing composed instruments in real-time. *Proc. of the 1998 IEEE Int. Conf. on Systems, Man and Cybernetics (SMC'98)*, pp. 1,080–4.

Wessel, D. 1979. Timbre space as a musical control structure. *Computer Music Journal* **3**(2): 45–52.

Wessel, D. 1991. Instruments that learn, refined controllers, and source model loudspeakers. *Computer Music Journal* **15**(4): 82–6.

Winkler, T. 1995. Making motion musical: gestural mapping strategies for interactive computer music. *Proc. of the 1995 Int. Computer Music Conf.*, pp. 261–4. San Francisco, CA: International Computer Music Association.

Winkler, T. 1998. *Composing Interactive Music. Techniques and Ideas using Max*. Cambridge, MA: The MIT Press.