Roel Vertegaal
Department of Computing
University of Bradford
Bradford, BD7 1DP, UK
roel@bradford.ac.uk

Ernst Bonis
Music Technology
Utrecht School of the Arts
Oude Amersfoortseweg 121
1212 AA Hilversum, The Netherlands

ISEE: An Intuitive Sound Editing Environment

This article presents ISEE, an intuitive sound editing environment, as a general sound synthesis model based on expert auditory perception and cognition of musical instruments. It discusses the backgrounds of current synthesizer user interface design and related timbre space research. Of the three principal parameters of sound—pitch, loudness and timbre—ISEE focuses on control of timbre, and affects only the range of pitch and loudness. Timbre is manipulated using four abstract timbre parameters: *overtones, brightness, articulation* and *envelope*. These abstract timbre parameters are implemented in different ways for different instruments. They define *instrument spaces* of which a hierarchy can be built to structure refinement of timbre parameter behavior. An Apple Macintosh implementation of ISEE is described.

ISEE has four main advantages over traditional sound synthesis editors. Firstly, it allows musicians to control sound synthesis as they control their musical instrument: by continuous movement, reducing cognitive control load. Secondly, it uses timbre parameters identified by human experience instead of indirect and intricate synthesis model parameters. Thirdly, it integrates a librarian system in the sound synthesis model. Finally, it enables transparent use of several synthesis models at a time.

Introduction

Creating high-quality sounds using a synthesizer is difficult to learn. We encounter some of those difficulties experienced by novice users each time a new model is introduced. Different implementations of the same synthesis model often use their own cryptic naming and parameter values, which lead to unnecessary communication problems.

The same communication problems arise when using different synthesis models. When a truly new synthesis model is introduced, manufacturers often implement the model's parameters directly as user interface parameters. The implementation of FM synthesis in the YAMAHA DX7 series is a good example. The DX7 was a success because of its highly cost-effective sound generation method. However, the user interface of this synthesizer is based upon control of an FM process, instead of a sound generation process. In fact, sound seems to be only a side effect of the FM process. Many novice users have the impression that creating sounds on an FM synthesizer is a stochastic process, which of course, is not the case.

This we see as one of the main reasons for the wide-spread use of (factory) presets by many musicians. This is sad because many of these people would rather use the full power of the synthesizer, yet find themselves in front of a device with a high learning curve. Though the problem is most acute with FM synthesizers, we feel that this issue is not intrinsic to FM, but mainly caused by the way the synthesis model is presented in current user interfaces.

Human-Synthesizer Interaction

Have user interfaces of synthesizers kept pace with sound technology? Large modular analog synthesizers had as many parameters as current synthesizers. Each parameter could be controlled directly with a knob. This, and the fact that modules had to be connected by wires, often led to an incomprehensible crisscross of knobs and wires. Miniaturization has made powerful desktop synthesizers possible, but increased the need to structure parameters by hiding them. Elaborate command structures were set up to enable the user to reach these hidden parameters. The benefit was that user interfaces became more hierarchically structured, thus making the synthesis process potentially easier to overview. However, at the same time it became increasingly difficult to directly reach a particular parameter.

State-of-the-art synthesizer models provide the user with a small bitmapped LCD display showing rather crude graphics and a menu and function-key based control system, which causes most control actions to be discrete. It is only once the targeted parameter is found that a slider or alpha-dial can be used to directly manipulate the parameter setting.

With the advent of MIDI editors, many of the acute overview and navigational problems, due to small displays and discrete controls, were solved by use of graphically oriented systems such as the Apple Macintosh as a front end to the synthesizer for editing and storage of sound patches. However, the almost 1:1 mapping of the user interface parameters with the synthesis model parameters was still maintained. We argue that these editors therefore do not fully comply with the principles of direct manipulation.

Direct Manipulation

Direct manipulation is a technique where objects and actions are represented by a model of reality. Physical action is used to manipulate the objects of interest, which in turn give feedback about the effect of the manipulation. A good example is transposing using a notation editor, in which case the metaphor is the note symbol, the action is moving the note vertically on the staff and feedback consists of note and hand position and the resulting audible change in pitch. Shneiderman (1987) argues that with direct manipulation systems, there may be substantial task-related semantic knowledge (e.g., the composer's knowledge about score writing), but users need to acquire only a modest amount of computer-related semantic knowledge and syntactic knowledge (e.g., the composer need not know that a score is not just put in a drawer, but in fact is saved as a MIDI file on a disk, nor that transposing consists of applying a *change-key-number* function to the *note on* and *note off* events of the note). To achieve maximum effect, computer-related semantics need to be replaced by task-related semantics.

Suppose we want to make a tone brighter. Using subtractive synthesis we could choose to manipulate the filter cutoff frequency, which directly affects the sound in the appropriate way. Using digital FM synthesis, one could choose to change the output level of a modulator. Though most of the time this seems to affect the brightness of the sound, in fact, one controls the width of the spectrum, which might result in noise due to aliasing if, for instance, operator feedback is active. Many parameters in FM synthesis have these intricate side effects on other parameters. A first step in making the user interface of a synthesizer more intuitive is to provide a more direct mapping between task related semantics (I want to make a sound brighter) and synthesizer related semantics (then I need to change the output level of the modulator or the feedback level or both). Can a synthesizer not have a brightness parameter?

A second step is to simplify syntax by reducing the number of actions needed to reach a goal, making the physical action more direct. The latter is one of the main reasons why editors are so much easier to use than built-in user interfaces.

Using the Motor System

Musicians usually have well developed motor skills, enabling them to control their instrument in a refined way. When a musician starts practicing a piece, he needs to adjust errors using feedback consisting of visual, auditory, tactile and muscular receptor information about the result of an action.

During the learning process, however, priority shifts from visual and auditory feedback to tactile and muscular receptor feedback, eventually resulting in the ability to perform without visual or auditory feedback (Keele 1973). An explanation for this phenomenon is the compilation of movements into motor programs (Keele 1968). According to Fitts and Posner (1967) linkage of motor programs during the final autonomous phase of skill learning reduces the amount of cognitive control necessary, clearing the mind for other tasks such as creative decisions. Musicians using modern synthesizers are often limited in their use of timbral expression during a performance. For each type of sound, different hardware parameters must be manipulated in a different way to achieve the same timbral goal. If in addition the synthesizer has a highly hierarchical and push-button controlled user interface, the sheer number of different discrete actions make it impossible to condition timbral manipulation. The more parameters needed and the more side effects each parameter has, the more difficult it becomes to reach the autonomous learning phase. We feel that this impairs the creativity of both synthesizer player and composer.

Timbre Space

Wessel (1974), Grey (1975) and Plomp (1976) proved it possible to explain differences in timbre with far fewer degrees of freedom than are needed by most synthesis algorithms. This implies that part of the solution for the timbre control problem lies in finding a suitable mapping between a low dimensional controller and the high dimensional synthesis algorithm.

Wessel (1985) suggested using multidimensional scaling techniques (Shepard 1974) to find such a mapping. He derived a timbre space from a matrix of timbre dissimilarity judgements made by humans comparing all pairs of a set of timbres. In such a space timbres that are close sound similar, and timbres that are far apart sound different. Feiten & Ungvary (1991) are making progress in training a neural network to automate the organization of sounds in a timbre space.

To use a timbre space as a synthesis control structure one specifies a coordinate in the space using an input device. Synthesis parameters are then generated for that particular point in space. This involves interpolation between the different originally judged timbres. A crude approach to implementing a timbre space for synthesis control would be to create a lookup table where for every coordinate a corresponding synthesis parameter set is defined which only needs to be looked up, providing a very efficient translation scheme. However, this approach claims considerable storage space, imposes problems on automated interpolation and therefore makes the definition task too laborious. Fortunately, more graceful methods have been found. Lee & Wessel (1992) report that they have successfully trained a neural network to generate parameters for several synthesis models with timbre space coordinates as input, automatically providing timbral interpolation. This approach however involves substantial computational power in order to train the neural network.

Plomp (1976) indicates that when using multidimensional scaling to define timbre spaces the number of timbre space dimensions increases with the variance in the assessed timbres. This makes it difficult to derive a generalized synthesis model from this strategy. When trying to reduce the number of dimensions artificially by using several less varied timbre spaces, the dimensions of the different timbre spaces might not correlate, which could cause usability problems if used as synthesis parameters.

Grey (1975) theorizes about the nature of the dimensions of the 3D timbre space he derived from an experiment in which 16 closely related re-synthesized instrument stimuli with similar envelope behavior (varying from wind instruments to strings) were compared on similarity. He indicates that one dimension could express instrument family partitioning, another dimension could relate to spectral energy distribution, and a third dimension could relate to the temporal pattern of (inharmonic) transient phenomena. Though these conclusions cannot simply be generalized, they do give us an indication of the nature of appropriate parameters to be used when generalizing timbre space as a synthesis model.

ISEE: The Intuitive Sound Editing Environment

An Overview of the ISEE Model

Wessel (1991) states that it is time for a higher level, synthesizer independent language. Similarly, Eaglestone (1988) relates the control problem to that of achieving data independence in a database environment, and hence achieving an abstract, user oriented interface. The **I**ntuitive **S**ound **E**diting **E**nvironment was designed to be just that. It is a synthesizer and synthesis model independent user interface designed to make use of typical musician's skills.

The principal concept of ISEE is encapsulation of synthesis expertise in the synthesis model. Four abstract timbre parameters were identified through qualitative observation of expert synthesis practice. Because of their high level of abstraction, these parameters have important orthogonal properties making them suitable as a basis for the high level ISEE synthesis model. The actual implementation of the abstract parameters depends on the required refinement of synthesis control. A scaled implementation of the four parameters is called an *instrument space.*

The term *instrument space* seems more appropriate than *timbre space* because an ISEE instrument space not only controls the timbre, but also defines the range and type of pitch and loudness behavior of the instrument(s) it encloses. However, explicit pitch and loudness controls are not included in the model since they are already incorporated in the controlling (MIDI) instrument.

The first two of the abstract timbre parameters relate to the spectral envelope and the last two to the temporal envelope: the *overtones* parameter controls the basic harmonic content; the *brightness* parameter controls the spectral energy distribution; the *articulation* parameter controls the spectral transient behavior as well as the persistent noise behavior; and the *envelope* parameter controls temporal envelope speed. The first three parameters are similar to those identified by Grey (1975).

A hierarchy of interconnected instrument spaces was devised to structure fine-tuned application of these abstract parameters for refined synthesis control. Since instrument spaces are ordered in the hierarchy according to their refinement, scale can be used as a hierarchy control structure. When a musician is interested in the sound of a particular instrument group in an instrument space, he can jump to a more refined instrument space filled completely by that sole instrument group by indicating the part of the instrument space of interest and asking the synthesis model to zoom in. Alternatively, when interested in a broader perspective of instruments, the musician can jump to a broader instrument space by indicating his wish to zoom out. More expert users can also make use of a traditional hierarchy browser, e.g., when constructing new instrument spaces.

A Taxonomy of Instrument Spaces

A categorization scheme was derived from expert analysis of existing instruments using think-aloud protocols, card sorting and interview techniques. The expert used this scheme to establish the necessary parameters to synthesize a target instrument. The categorization method was used to set up a taxonomy of instruments based on expert perception and cognition, and is incorporated in ISEE to structure the instrument space hierarchy.

Figure 1 depicts a partial taxonomy of instrument spaces matching the expert categorization scheme. The first criterion is the temporal envelope model, the second criterion is the harmonicity of spectrum, on which further categorization depends. For harmonic instrument spaces, further classification lies in transient behavior and formant structure (the latter is not included in the partial taxonomy), since those are important properties when distinguishing between harmonic timbres. For (decaying) inharmonic instrument spaces, vibrating body type needs to be established first. Further structuring of both harmonic and inharmonic instrument spaces into instrument families can be done according to the Sachs-Hornbostel classification system.

Since this taxonomy relates more directly to the perception and cognition of sounds by a trained listener, it gives better direction for classification of (electro-)acoustic sounds than the Sachs-Hornbostel system.

Instrument Space Layout

The layout of an instrument space very much depends on its refinement and the instrument group(s) it encloses and is defined by a specific implementation of the abstract synthesis parameters and constant properties of that space.

Constants of an instrument space are all synthesizer parameters that need to be set up for the timbre parameters to work. They include instrument-specific tuning, algorithm selection, voice name, etc. Generally, timbre parameter functionality and instrument space hierarchy arrangement are mapped using the following heuristics: from low to high, from harmonic to inharmonic and from mellow to harsh. The *envelope* parameter functionality is mapped from fast to slow.

To be able to define a point in an instrument space by combining the data of its projections on the four axes, i.e., the four timbre parameters, it is important to keep the timbre parameters as orthogonal as possible, not an easy task when defining an instrument space for an FM synthesizer. It is best to look at the instrument space taxonomy in figure 1 to explain timbre parameter implementation. In the root space, the *envelope* parameter is used to decide whether the envelope model is sustaining or decaying. In this space the attack will be set at a constant rate, sufficiently short to fall within most instrument families' range.

Fig. 1. An example partial taxonomy of instrument spaces.

Fig. 2. The ISEE system connections in the MIDIManager Patchbay and a diagram. Control Monitor (1) sends coordinate keys through a pipe (2) to the Interpreter (3) which pipes the corresponding synthesizer commands (4) to the MIDI output port (5).

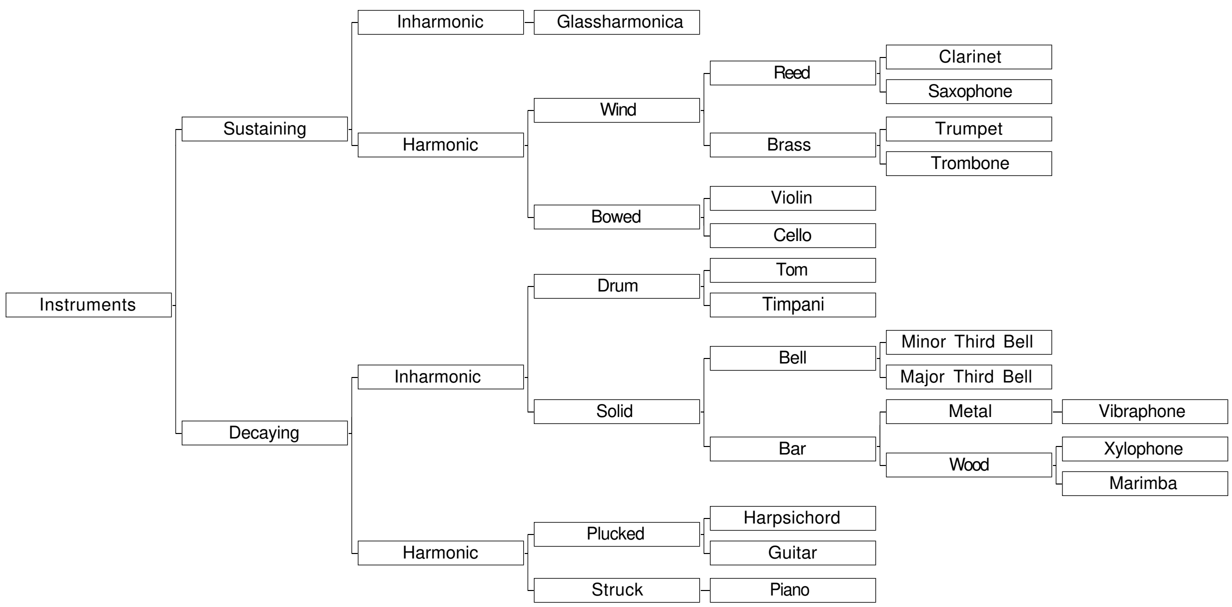

Fig. 1

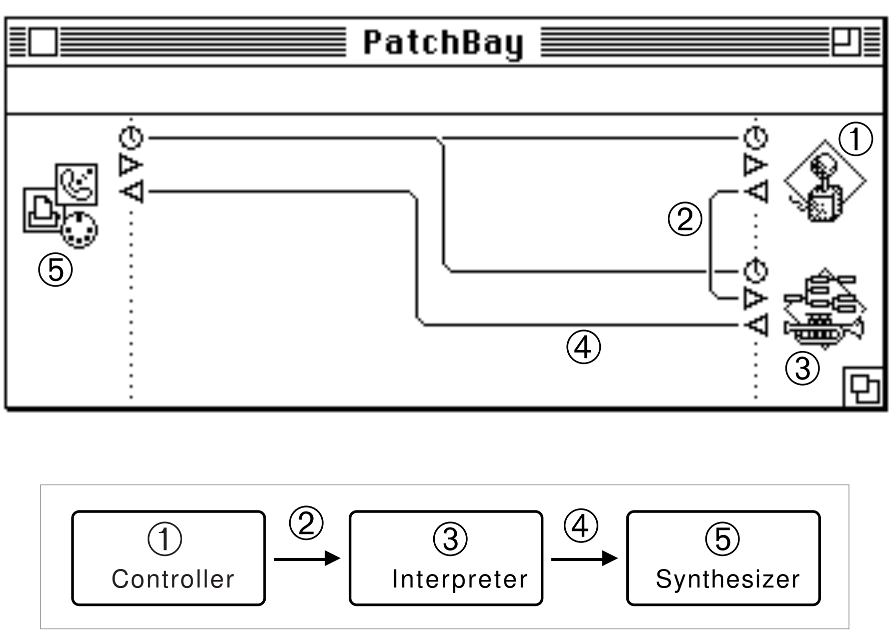

Fig. 2

Fig. 3. The Control Monitor application is used to control and monitor the position in the hierarchy (depicted by the middle icon) and the position in the current instrument space (indicated by the two dots). Two buttons are used to zoom out to the parent space (Harmonic) or zoom in to the child space (Violin) closest to the 4D position indicated by the dots.

Fig. 4. A sample hierarchy file in Interpreter, with its instrument taxonomy browser in the background. The instrument space definition window shows how the overtones parameter is defined by multiple layers of MIDI blocks. In front, the hex edit window shows the system exclusive data contents of the block covering row 1, setting 16-32 of the overtones dimension of the current space.

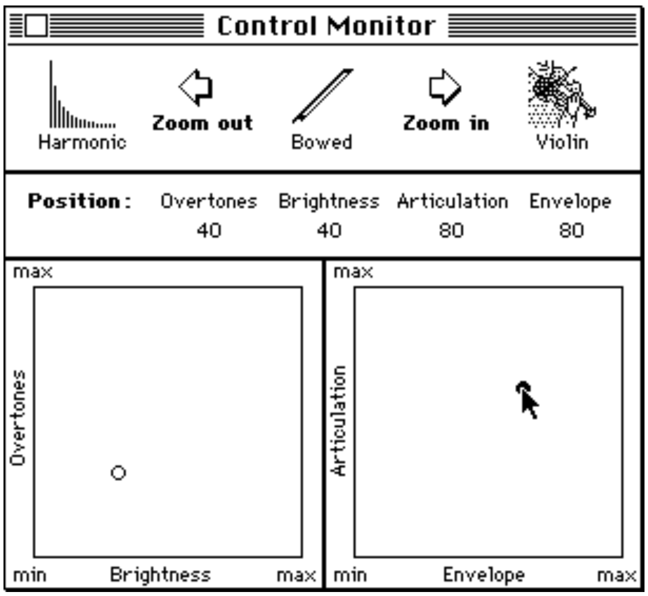

Fig. 3

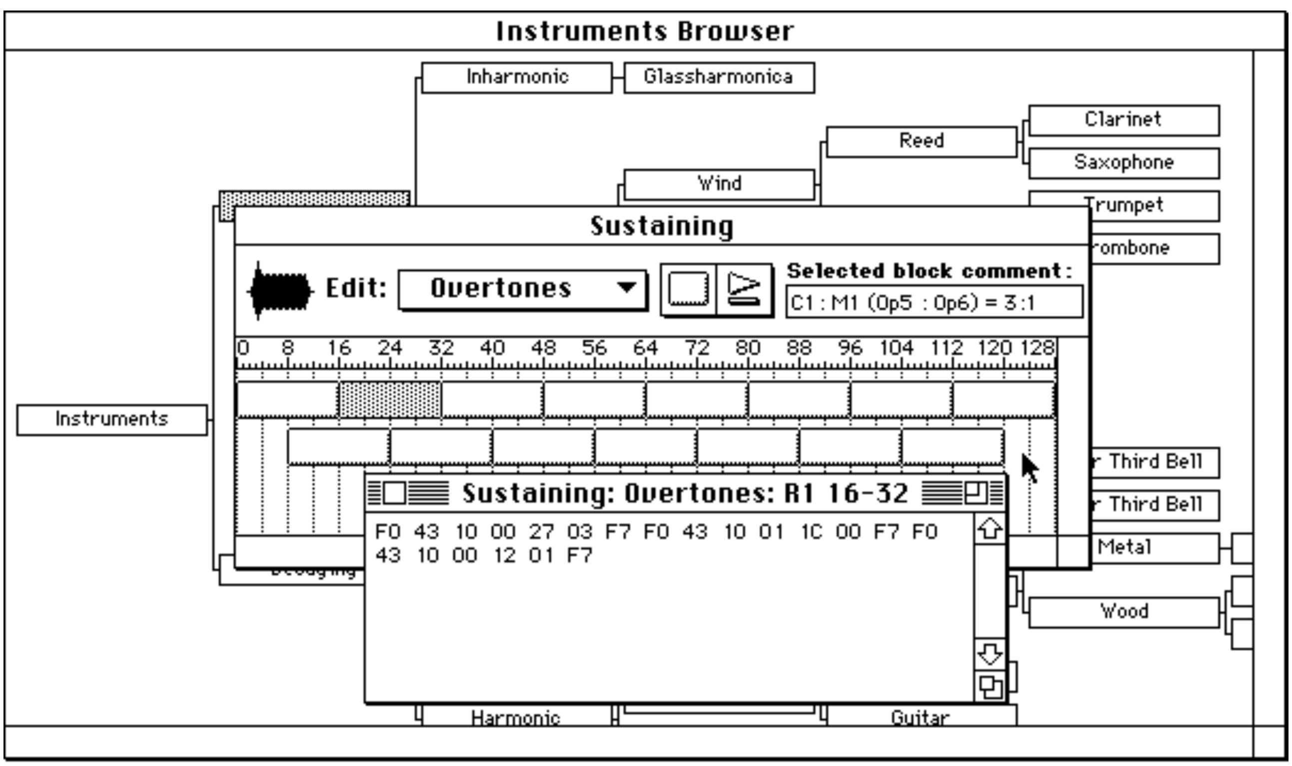

Fig. 4

The envelope model selection will limit further behavior of the *envelope* parameter down the hierarchy. If sustaining is selected, the *envelope* parameter will be limited to change the attack, only slightly adjusting the rest of the temporal envelope. If decaying is selected, this parameter will be implemented to affect the decay.

Selection is done by moving the *envelope* parameter towards the targeted envelope model using auditory and visual feedback, and zooming in, e.g., by pressing a zoom button. In the root space, the *overtones* parameter affects basic harmonic content from harmonic through harmonic with formants, odd harmonic and inharmonic to noise. The *brightness* parameter will affect the bandwidth emulating basic filter behavior, from low pass through all pass to high pass. Finally, the *articulation* parameter will affect the balance between the rise time of the lower partials and the higher partials, from the typical brass transient where lower partials rise first, to a string transient emulation where higher partials come first, the ultimate sound depending on the musical use of the wind or keyboard controller.

The decisive parameter(s) for hierarchy traversal changes per level, e.g., one level down in the hierarchy the *overtones* parameter is decisive, another level down, the same parameter decides between *Solid* and *Drum* in the inharmonic decaying instrument space, and a combination of the *overtones* and *articulation* parameter decides whether the harmonic sustaining instrument space will be refined into *Wind* or *Bowed.*

If we look at the layout of the *Brass* instrument space in figure 1, the *overtones* parameter is used to distinguish the different instruments registers from low to high, the *brightness* parameter acts as a low-pass filter, the *envelope* parameter affects attack speed and the *articulation* parameter affects the amount of roughness during the attack, relating to the amount of hiss during the steady state as well. The relation between the rise of lower and higher partials has now become part of the constant behavior for this instrument space, and is defined by the constants of the instrument space.

If we look at the most refined instrument space level in our partial taxonomy, e.g., the *Violin*, we see that the *overtones* parameter can be used to describe the relation of the bow to the bridge, from *flautando* to *sul ponticello.* Here, the *brightness* parameter relates to the bow pressure on the string, the *articulation* parameter controls the harshness of the inharmonic transient components and the *envelope* parameter controls the attack speed. This scheme of refined implementation of abstract timbre parameters gives ISEE potential as an intuitive general controller for physical modeling.

The ISEE Implementation

A first prototype of ISEE was developed in 1990 to test the validity of the abstract timbre parameter paradigm (Vertegaal 1992). The next paragraphs describe the forthcoming upgrade which facilitates instrument space definition and incorporates hierarchical structuring. ISEE runs on any Apple Macintosh with Apple MIDI Manager and System 7. It requires a minimum of one megabyte of free memory. ISEE consists of two module applications: the *Control Monitor* application, used to control and monitor the positioning within the current instrument space and within the instrument space hierarchy, and the *Interpreter*, which translates user interface control data into synthesizer control data using a database of recorded synthesizer commands. Figure 2 shows the two applications' communication link in the MIDI Manager Patchbay application, which provides piping of MIDI data between applications on a Macintosh by means of software-emulated MIDI cables.

Control Monitor

The Control Monitor window is depicted in figure 3. A mouse is used to position two points in two coordinate systems: the first defined by the *overtones* and *brightness* parameters, the second by the *articulation* and *envelope* parameters. Zooming into and out of a region is done by pressing the corresponding buttons. The left icon depicts the parent instrument space, the middle icon the current instrument space and the right icon the instrument space to which the system will jump at *zoom in* command. The Control Monitor application can easily be linked to specific hardware controller drivers, in which case it can be used to monitor location in the instrument space and the hierarchy.

A sequencer or *Max* can be used to record and alter ISEE Control Monitor data, e.g., to establish wave sequencing on any synthesizer capable of real-time synthesis parameter control using MIDI.

Interpreter

The Interpreter translates the 4D locations it receives from the Control Monitor to corresponding MIDI synthesizer parameter data. It incorporates an instrument space classification browser (see figure 4), which provides direct selection of instrument spaces and tools to create new spaces, connect and edit them. When the Interpreter receives a *zoom in* command it will respond by looking up the child instrument space located nearest to the current position and it will jump to that space, broadcasting new constant parameter settings to the synthesizer and moving the current position to a spot in the child space that provides smooth transition.

The Interpreter incorporates an instrument space editor (see figure 4), providing MIDI data recording from external sources such as an Opcode *Galaxy* editor as well as manual hexadecimal input, to define each timbre parameter and the constant behavior of an instrument space. After a timbre parameter has been selected, its 128 positions can be defined using multiple layers of *MIDI blocks*. One block groups all necessary MIDI synthesis parameter commands (i.e., system exclusive commands) to make one conceptual change in timbre with the specified timbre parameter. Blocks can be labeled to provide comment about their functionality. Brick layering of blocks can be used to cross-fade between synthesizer timbre changes. For instance, one definition of the *brightness* timbre parameter for a particular instrument space could incorporate several implementations for different synthesis and synthesizer models at a time. The DX7 implementation could change the modulator output level from 10 to 80 on the first row. An incorporated SY 77 implementation might use the second row to change the filter cutoff frequency from 0 to 127, thus providing similar functionality. MIDI blocks incorporate a real-time interpolation feature to facilitate definition.

Future Directions

ISEE is a next step in the development of less complicated tools for creative sound synthesis by musicians. Many assumptions made in the design are based upon expert opinion, partly because many of the functions of human timbre perception and cognition are still unknown. Qualitative testing of ISEE by musicians with less experience in sound synthesis is a step to be taken in the near future. The hardware controller (currently a mouse) needs investigation. An absolute controller is preferred, since it enables musicians to use motor system memory as they are used to, but problems such as nulling the device (Buxton 1986) when entering a different instrument space need to be solved first.

Different graphical representations of timbre space control need to be developed and tested. Experiments in these directions involving a sample of music students are being set up at the Department of Computing at the University of Bradford. Instrument space layout and hierarchical classification need further investigation and implementation in order for ISEE to be released as a general sound synthesis system. Our ultimate goal is implementation of ISEE as a built-in intuitive synthesizer user interface, with mounted hardware controllers to provide ISEE's functionality on stage, and a computer front-end for instrument space editing.

The Effective Dimensions of Instruments

An interesting view upon musical instrument perception arises from the ISEE hierarchy of connected instrument spaces. The effective dimensions (the efficiency with which something fills space) of an instrument change from instrument space to instrument space, analogous to the way one's visual and auditory perception of an orchestral instrument in a concert hall changes if one moves from a balcony seat to an orchestra seat. The effective dimensions of an instrument in our hierarchy can vary from zero, filling just one point in 4D space in a large scale instrument space, to four, filling a whole refined instrument space, with the possibility of fractional effective dimensions (Mandelbrot 1977) between these extremes. The fractal nature of scaling instrument spaces calls for further research.

Conclusion

In this article, we have identified problems that could occur when using current sound synthesis user interfaces. We have discussed inconsistencies in the current synthesis editors' implementation of direct manipulation. The timbre space paradigm has been identified as the cornerstone of a next generation synthesis model, featuring more direct timbre manipulation. The Intuitive Sound Editing Environment was introduced as such a next generation sound synthesis model. We end this article by discussing the advantages and disadvantages of the ISEE approach.

The reduction of timbre parameters to four enables the use of absolute input devices that allow musicians to control sound synthesis as they control their musical instrument: by continuous movement. It was shown that this use of motor system control leads to a reduction of cognitive control load, and we argue that the ISEE user interface will enable the musician to focus attention on creative design. Furthermore, less synthesis model specific knowledge is needed when manipulating timbre using ISEE. The built-in instrument taxonomy gives users a model for structuring sound patches in files.

The ISEE timbre parameters were identified through human expert timbre cognition and perception, instead of dictated by the synthesis model parameters. ISEE offers a synthesis model independent language which enables transparent use of multiple synthesis models at a time and reduces the musician's problems of adaptation to new synthesis models or synthesizer models.

ISEE extends synthesis model patch programming to include the definition of four timbre parameters. Instrument space development will remain a human task in the near future, making redefinition for new synthesis models an even more elaborate effort than preset synthesizer patch development already was.

Limiting user control over hardware parameters to create a more user friendly interface has always come with a cost. However, use of conventional patch editing software remains an option for the expert. Let us conclude by emphasizing that ISEE is a low cost, synthesis model independent approach, bringing intuitive sound editing in the form of instrument space navigation to the homes of musicians.

Acknowledgements

We would like to thank Dick Rijken, John Chowning, Adrian Freed, Richard Boulanger, S. Joy Mountford, Tamas Ungvary and Barry Eaglestone for their valuable insights, directions and support. Thanks to Hendrik Jan Veenstra and Albert Verschoor for their essential work during the knowledge acquisition phase. Thanks to Deborah Twigger for proofreading. We would further like to thank the Center for Knowledge Technology and Iain Millns for providing the necessary facilities.

References

Buxton, W. 1986. "There's More to Interaction than Meets the Eye: Some Issues in Manual Input." in D. A. Norman and S. W. Draper, ed. *User Centered System Design: New Perspectives on HCI*. Hillsdale, NJ, Lawrence Erlbaum Associates: 319-337.

Eaglestone, B. 1988. "A Database Environment for Musician-Machine Interaction Experimentation." *Proceedings of the 1988 ICMC*, Cologne, International Computer Music Association.

Fitts, P. and Posner, M. 1967. *Human Performance*. London, Prentice-Hall, Inc.

Feiten, B. and Ungvary, T. 1991. "Organisation of Sounds with Neural Nets." *Proceedings of the 1991 ICMC*, Montreal, International Computer Music Association.

Grey, J. 1975. "An Exploration of Musical Timbre." Ph.D. Dissertation, Dept. of Psychology, Stanford University. CCRMA Report STAN-M-2.

Keele, S. 1968. "Movement Control in Skilled Motor Performance." *Psychological Bulletin* 70: 387-402.

Keele, S. 1973. *Attention and Human Performance*. Pacific Pallisades, Goodyear Publishing Company.

Lee, M. and Wessel, D. 1992. "Connectionist Models for Real-Time Control of Synthesis and Compositional Algorithms." *Proceedings of the 1992 ICMC*, San Jose, International Computer Music Association.

Mandelbrot, B. 1977. *The Fractal Geometry of Nature*. New York, W. H. Freeman and Company.

Plomp, R. 1976. *Aspects of Tone Sensation*. London, Academic Press.

Shepard, R. 1974. "Representations of Structure in Similar Data: Problems and Prospects." *Psychometrica* 39: 373-421.

Shneiderman, B. 1987. *Designing the User-Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA, Addison Wesley.

Vertegaal, R. 1992. “ISEE: ontwerp en implementatie.” Music Technology Dissertation, Utrecht School of the Arts, The Netherlands.

Wessel, D. 1974. “Report to C.M.E.” University of California, San Diego.

Wessel, D. 1985. “Timbre Space as a Musical Control Structure.” in C. Roads and J. Strawn, ed. *Foundations of Computer Music*. Cambridge, MA, MIT Press.

Wessel, D. 1991. “Let’s Develop a Common Language for Synth Programming.” *Electronic Musician* 1991(8): 114.