Proceedings of The 1980 International Computer Music Conference

"OBJED" and the Design of Timbral Resources

by

W. Buxton, S. Patel, W. Reeves,
and R. Baecker¹

Structured Sound Synthesis Project (SSSP),
Computer Systems Research Group,
University of Toronto,
Toronto, Ontario,
Canada,
M5S 1A1

1. INTRODUCTION

One of the main attractions of electroacoustic music is the potential for composers to design and control their own palette of timbral resources. A key frustration, however, is due to the difficulty in actually doing so. Consequently, in spite of technological advances, systems appearing on the market seem to be reverting to a reliance on preset timbres which is reminiscent of organs and conventional instruments. Very little progress seems to have been made in the development of tools which aid composers in "rolling their own" timbres.

With analogue equipment the musician could use a "hands-on" approach to exploring the timbral potential of various configurations. The adoption of digital technology provided greater precision, sophistication, and potential, but initially made timbral exploration more remote. With the current trend towards highly interactive real-time digital systems, there is no reason why the composer should not now have the best of both worlds. It is towards such an end that this paper is directed.

Ideally, timbre should be controlled according to perceptual rather than acoustical attributes. However, our limited understanding of perception, cognition, and acoustics makes this difficult at present.² If we must deal with timbral specification in mainly acoustic terms, it is important to provide an environment for doing so which minimizes the non-musical problems of the task, and which permits the composer to develop an ability to understand and predict the perceptual consequences of changing acoustical parameters. The realization of such a timbre editing environment requires:

- that the data being edited is clearly represented.
- that there is a good correspondence between the things that the composer wants to do and the operators that the editing environment provides.
- that the burden on the user in undertaking book-keeping and administrative tasks is minimal.
- that the syntax for all transactions is succinct, intuitive, and self-consistent.
- that the system have a high tolerance for user error, and encourage exploration and experimentation: that is, learning.

1. Patel is currently with Human Computing Resources, Toronto, Canada. Reeves is now with Lucas Film Corp., Novato, Cal. Baecker is with both U. of T. and Human Computing Resources.

2. However, work by Grey (1975, 1977) and Wessel (1980) is making considerable headway in this direction.

Proceedings of The 1980 International Computer Music Conference

We now examine one program which attempts to fulfill these requirements. In so doing, the intention is to provide a case study which serves as the basis for a discussion of the issues involved. It is hoped that as much will be learned from the program's deficiencies as from its strengths.

2. OBJED AND OBJECTS

Objed is a program which permits named sets of timbral characteristics (called "objects") to be defined, auditioned, and modified.³ Objects have three properties:

- each is contained in a separate, uniquely named file.
- each defines a set of timbral characteristics (including time-varying functions) which can be used to orchestrate one or more notes.
- the parameters determining pitch, maximum amplitude, and duration are determined by the orchestrated note rather than the object itself.

Objects can be compared to MUSIC V instruments (Mathews, 1969), or "patches" on analogue synthesizers. The concept is general, but the current SSSP implementation restricts the composer to working with a limited set of object types. Type is determined by the method of sound synthesis employed from among fixed waveform, frequency modulation (Chowning, 1973), additive synthesis (Moorer, 1977), VOSIM (Kaegi & Tempelaars, 1978), and waveshaping (Arfib, 1978; Le Brun, 1978). The main consequence of this restriction is that the composer's task becomes one of plugging values into a fixed template. As a result, one works at a level halfway between the two extremes of instrument definition at the unit-generator level, and the use of presets.⁴

Ideally, all three levels of working should be available in a complete system. However, since the implementation functions at the middle level, so do the examples. That is not to imply that the techniques used are incompatible with working from a lower level. It is important to separate the issue of the *means* of editing from the *content* (what is being edited). Just as the same program is used to edit objects of all types, so could the same concepts be applied to MUSIC V instruments. Furthermore, combined with presets, new objects can be "cloned" from existing ones, where the composer uses the preset closest to his objective as a point of departure.

3. OBJECT EDITING

On initial entry, *objed* assumes the state shown in Figure 1. The first thing to notice is how the display is divided into five regions. Different types of functions are located in particular regions. The main region, which occupies the upper two thirds of the screen, is where the actual data being edited is displayed. In the example we see that a simple fixed waveform object is being edited. Its components, all of which are graphically displayed, are: a waveform, a time varying function controlling pitch, and an envelope controlling the contour of the note's volume.

To the left, below the main region, is an area containing data pertaining to the pitch, volume, and duration at which the object being edited can be auditioned. Remember, however, that these values are not part of the object. They are simply conveniences for exploring its behavior in different contexts.

The central panel in the lower part of the screen contains various options which allow the user to change the state of the editing environment. Options available include changing the type of object being edited, and changing the way in which the edited object can be auditioned.

3. The notion of object used here derives from that developed by Truax (1976) in the POD programmes.

4. Unit generators are signal processing elements which are the lowest level module used in defining MUSIC V instruments.

Proceedings of The 1980 International
Computer Music Conference

Figure 1. Objed on Initial Entry

The panel to the lower right is dedicated to the saving and retrieving of objects with a minimum of effort. Finally, the elongated region along the bottom left edge is a "window" which permits the composer to access the outside world without leaving *objed*.

The method of working with the program is designed to minimize the amount of rote learning which must be undertaken by the user. Rather than memorize a large number of commands, the user need only remember a simple strategy which forms the basis for all interactions: when you want to change something, just point at the diagram or word which represents it on the display, and depress the selection ("Z") button on the tablet's cursor. Any consequent options will then be presented, and the same method of interaction is applied to them.

It is often useful to begin by auditioning the displayed object. This establishes a frame of reference for future changes. The object is played by activating the word, or *light button*, PLAY seen in the lower part of the working area. On so doing, the sound is

Proceedings of The 1980 International
Computer Music Conference

immediately heard. Remember, though, that the object itself has no specific pitch, duration, or maximum amplitude associated with it. These must be provided externally for the object to be heard. When PLAY was activated, the sound had a pitch of A4, dynamics of about *mf*, and a duration of 1/4 note at mm = 60. Obviously it is desirable to be able to change these values so as to learn more about the object's behavior. This is the function of the lower-left sub-panel seen in Figure 2.

Figure 2. Graphical Pots for Performance Parameters

Here are seen three "graphical potentiometers", one for each performance parameter. The value for each parameter is displayed numerically in the boxes above, and its relative value is seen by the position of the potentiometer's "handle". The value of any pot can be changed by "dragging" its handle up or down using the cursor. Alternatively, the user may point at the box above the pot, activate the Z-button, and type in a numerical value. In this case, the tracking symbol becomes an icon of a terminal (as seen in Figure 3), as a prompt that something must be typed.

Figure 3. Setting a Graphical Pot by Typing

Proceedings of The 1980 International Computer Music Conference

There are several points to note in what has been seen thus far. First, *nothing* has been defined from scratch. On entering the program the user could immediately audition a reference sound, and begin working from there. A basis for learning is provided, even for the user who does not know what a sine wave is. Second, parameters such as pitch and volume are changed using a technique which has an analogy in most user's previous experience. Finally, iconic prompts are provided at the point of focus of the user's attention, as was seen with the "terminal" prompt in the last example.

Let us now consider changing one of the attributes of the object itself. As an example, let us alter the waveform associated with the object. Pointing at the picture of the current waveform and depressing the Z-button will cause the panel seen in Figure 4 to appear.

Figure 4. Waveform Selection

A menu consisting of the eight waveforms currently loaded in the synthesizer appears down the right margin of this panel. Selecting one of these waveforms with the cursor

and then activating the light-button DONE will return us to the original panel of Figure 1. with the one difference that a drawing of the selected waveform will have replaced that of the sinusoid. This brings up an important point: the current state of the object being edited is always clearly displayed, thereby eliminating any mental burden of remembering its current attributes.

Returning to the previous example, if we wanted to define a new waveform rather than use one of the ones in the synthesizer, we could have done so. By activating the light-button DEFINE BY SPECTRUM (Figure 4), the panel shown in Figure 5 would have appeared.

Figure 5. Defining a Waveform by Spectral Content

What is seen in the work area is a bar-graph, where the height of the bars represents the relative amplitude of the harmonics of a sound. Harmonics one through sixteen appear from left to right. Any bar will jump to the height of the cursor when it passes over the bar with the Z-button depressed. Thus, a simple hand gesture allows the

spectral envelope of a waveform to be sketched. More importantly, the waveform is being synthesized all the while, and the user can use the graphic potentiometers in the lower left-hand panel to adjust the overall amplitude and the frequency of the fundamental. (Note how the placement and use of the latter is consistent with the main panel, thereby simplifying learning of the control structure.) Throughout, the spectral content of the sound and the graphics display are updated in real-time. The result is that musicians can quickly develop a sense of the perceptual effect of spectral content on steady-state tones.

Once the desired spectrum is defined, the composer activates the CALCULATE light-button, and the new waveform is calculated⁵ and displayed (in the time domain), as seen in Figure 6.

Figure 6. Waveform Defined by Spectral Content

The waveform is then loaded into the synthesizer and the user returns to the main panel by activating the command DONE. The new version of the object can then be auditioned, and other changes can be made. For example, the envelope associated with the object can be redefined using techniques comparable to those seen with the waveform.

Thus far, the object being edited has been of the type *fixed waveform*. We can experiment with another mode by activating the light-button FIXED WAVEFORM seen in the lower central panel (Figure 7). As a result, the panel will switch to present the set of object type options. Selecting "Frequency Modulation" will result in the display appearing as seen in Figure 8. The panel is the same as that seen in Figure 1, except that there are five additional elements in the working area. There is now a waveform displayed for both the carrier and modulating waves. In addition, graphical potentiometers are provided to set both the maximum index of modulation and the c:m ratio. Finally, there is a third time-varying function which controls the evolution of the index of modulation.

The example illustrates two points. First, the environment is the same regardless of object type. Second, the work panel previously seen is a subset of the current one. The benefit resulting from this self-consistency is that skills learned at a simple level can be applied to more complex tasks. Thus, the basis for good pedagogical practice is provided.

During an editing session, it is often desirable to be able to audition the object repeatedly. One way to do this is to activate the SINGLE button at the bottom of the

5. In fact, activating CALCULATE presents the user with the option of having the waveform calculated according to the weighed sum of the Fourier series, or the Chebychev polynomials. In each case, the bar heights represent the relative weights of the different order functions. The Chebychev polynomials are used to generate transfer functions to be employed in waveshaping synthesis.

Figure 7. Editing Environment Control Sub-Panel

environment control panel (Figure 7). The button will be renamed CYCLE, indicating that when PLAY is activated the sound will play repeatedly until killed by the user.

The problem still exists, however, that all of our interactions with the object's data take the two-part form: change a value, listen to the effect. What would often be more useful, when setting the index of maximum modulation for example, would be to emulate the operation of an analogue synthesizer. That is, it would be useful to turn the sound on in its steady state, and hear the effect of adjusting parameters while they are being changed. (This is the same thing which we have seen when defining waveforms by spectrum.) This can be accomplished by activating the CYCLE button, which then switches to STEADY. Activating PLAY will cause the object to sound, and the effect of adjusting any of the graphic potentiometers (including those affecting pitch and volume) will be heard immediately. When desired, we can return to the original SINGLE mode by activating the button STEADY. Thus, we have not only seen some new modes of auditioning objects, we have also introduced a new concept: that of the rotary switch. By convention, any light-button parentesized by the "*" character functions like a rotary switch. Activating the button will cause it to change to a new label. Repeated activation will cause it to rotate through its "cycle" and return to its original value. Each such button, then, provides a means of selecting from a set of options for a particular function. The mode currently in effect is that displayed. A means is provided, therefore, of accessing significant complexity, while still being able to hide it until the user is ready for it.

In spite of the flexibility just seen in auditioning an object, there is still one fundamental problem. The effect of hearing an object in isolation is often (usually?) very different than hearing it in some musical context. So far, we have only been able to hear an object as a single note. This can be altered by activating the NOTE MODE button (Figure 7). The result will be that the button is renamed SCORE - UNIFORM ORCHESTRATION, and the sub-panel controlling note parameters is switched, as seen in Figure 9. The box that appears can be thought of a "window" which looks out on all of the scores which the composer has created. The name of each score is listed in this window, and if there are more names than will fit, a method is provided to "scroll" through them. The point to note is this: when the PLAY button is activated, the score whose name appears between the horizontal lines of the window is performed. More important to our purposes, the score is temporarily orchestrated with the object currently being edited! Furthermore, a metronome marking controlling the tempo

Figure 8. Editing an FM Object

of the performance can be controlled by adjusting the graphical potentiometer beside the window. Pointing at any other name in the window will cause it to move between the horizontal lines so that it, too, can be played. Finally, we can "rotate" the mode switch one position further and change it to read SCORE - ORIGINAL ORCHESTRATION. In this case, when the score is played it retains its original orchestration. However, all notes orchestrated with the object whose name corresponds to that of the object being edited will be performed with the working copy of that object.

As a result of these assorted performance modes, the material being edited can be heard in a variety of musically useful contexts, thereby providing the composer with valuable tools to learn about the musical properties of the material being designed.

Before an object can be used outside of the context of the editing environment, it must be named, and then saved. An object's current name appears in the heavily outlined box at the top of the environment control sub-panel (Figure 7). Names can be

Figure 9. Auditioning Objects in the Context of a Score

changed the same way as anything else; you point at it, activate the Z-button, and type in the new name in response to the "terminal" prompt. Saving an object for future use is equally straightforward: the light-button SAVE seen in the bottom right-hand sub-panel is activated. This same sub-panel also provides the mechanism for retrieving previously defined objects. The technique used is one already seen, the use of a directory window. In this case, instead of looking out onto previously defined scores, the window provides selective vision upon objects, only. The name between horizontal lines is always the same as that of the object being edited. Selecting a different name causes that object to be displayed in the work area for purposes of editing or audition. One interesting option in this process has to do with the domain over which the window looks. That is, the window will either let us view objects which have been defined and stored on disk, or, those which are in primary memory having been worked on in the current session. Thus, while only one object can be edited at a time, several "working" objects can be kept in primary memory during a work session. Nothing need be saved unless it is desired for a future session.

The value of providing easy access to a set of working objects can be seen in our next example. One function which we feel is important to provide is the ability to compare objects in rapid succession. In order to do so, the data for each object to be auditioned must already be in primary memory. Even then, if we have to point to the object name, wait for its data to be displayed, and then move the cursor to activate the PLAY button, we have lost the instantaneous response which we desire. As a result, another light-button, COMPARE, is provided in the environment control area. Activating this button causes almost everything except the object window to disappear from the display. In this mode, objects are played immediately upon having their name selected with the cursor. Since they are already in primary memory, hand movement is minimized, and the object data is not displayed, rapid comparisons are possible.

The directory windows seen in the previous examples are the most important feature of the system in terms of one of our initial demands: minimizing the cognitive effort required for administration and book-keeping. In computer science terms, this largely translates to file input and output. The windows allow files to be input without having to type names, remember spelling, or having to extract the files of a particular type from among all those of different types in the file directory.

4. CONCLUSIONS

We have examined a particular program as a means of improving our understanding of how to provide composers with better control over timbral resources. The examination has concentrated more on the *means* than the content of the editing process.

Proceedings of The 1980 International Computer Music Conference

Some of the guiding principles which were of value in the design process were: self consistency among interactions, flexibility in modes of operation (such as in audition), clarity of presentation of data and operators, and the use of defaults.

While our experience with *objed* has been favourable, there are still some fundamental problems. Musically, the most important is this: designing a system around a score editor on the one hand, and an object editor on the other implies that composition can be partitioned between the sonological and deep-structural levels. This is a large assumption, and one which is clearly not justified in some musics. A more specific but related problem concerns the program's weakness in dealing with timbres which are aggregates of two or more objects functioning together. Finally, defining objects in terms of data plugged into a template leaves very little room for the specification of how the object adapts its behavior in particular contexts. How can the specification of procedural information (such as volume varying with pitch) be combined with the environment described?

Clearly, the issue of timbral specification and control is one which still requires a great deal of research. Hopefully the work presented here will help in bringing such control to the composer.

5. ACKNOWLEDGMENTS

The results reported in this paper have benefited from the input of numerous composers who have worked on the SSSP system, especially James Montgomery and Philippe Menard. As well, several conversations with David Wessel have strongly influenced our approach. Finally, Bob Pritchard and M. R. Lamb have made many helpful comments during the preparation of this manuscript.

The research reported in this paper has been undertaken as part of the SSSP of the University of Toronto. This research is supported by the Social Sciences and Humanities Research Council of Canada. This support is gratefully acknowledged.

6. REFERENCES

- Arfib, D. (1978). Digital Synthesis of Complex Spectra by Means of Multiplication of Non Linear Distorted Sine Waves. *AES Preprint No. 1319 (C-2)*.
- Chowning, J. (1973). The Synthesis of Complex Audio Spectra by Means of Frequency Modulation. *Journal of the Audio Engineering Society* 21: 528-534.
- Grey, J. (1975). *Exploration of Musical Timbre*. Stanford Univ. Dept. of Music Tech. Rep. STAN-M-2.
- (1977). Multidimensional Perceptual Scaling of Musical Timbre. *Journal of the Acoustical Society of America* 61: 1270 - 1277.
- Kaegi, W. and Tempelaars, S. (1978). VOSIM-A New Sound Synthesis System. *Journal of the Audio Engineering Society* 26: 418-424.
- Le Brun, M. (1978). Digital Waveshaping Synthesis. *Journal of the Audio Engineering Society* 27: 250-266.
- Mathews, M. V. (1969). *The Technology of Computer Music*. Cambridge: MIT Press.
- Moorer, J. A. (1977). Signal Processing Aspects of Computer Music - A Survey. *Proceedings of the IEEE*, 65: 1109-1137.

Proceedings of The 1980 International
Computer Music Conference

Truax, B. (1976). A Communicational Approach to Computer Sound Programs. *Journal of Music Theory* 20.2: 227-300.

Wessel, D. (1979). Timbre Space as a Musical Control Structure. *Computer Music Journal* 3.2: 45 - 52.