Patchmix and StochGran
Two Graphical Interfaces

Mara Helmuth
Columbia University Music Department
Dodge Hall, New York, New York 10025
mara@woof.columbia.edu, th-mhb@cs.columbia.edu

ABSTRACT: Two graphical interfaces for computer music composition deal with different methods of timbral synthesis. The first allows the user to build a patch of various unit generators and the second is a granular synthesis instrument. Patchmix is an instrument builder for the Cmix programming language. The user constructs a flow chart of unit generator icons and then compiles and tests the instrument. The code for the instrument is written out for later use and modification if desired. Constructing an instrument graphically allows for ease and spontaneity in the creative process. StochGran is a granular synthesis NextStep program. The user specifies each event in terms of grain rates, durations, locations, and frequency characteristics. These parameters may change over the course of the event and vary by a specified amount of deviation. The events are heard, modified, and saved within the interface. Patchmix was used to compose "Dragon of the Nebula", for computer generated tape. Both StochGran and Patchmix will be demonstrated on a NeXT machine.

Introduction

Patchmix and StochGran are two tools designed to facilitate computer music composition. One of the biggest problems when composing computer music is the amount of time spent in typing in lines of data or programming each subtle change in timbre. Often the flow of musical ideas is hampered by these tasks. These two programs make creating music with software synthesis somewhat easier. With Patchmix one can quickly construct icons graphically to generate a Cmix instrument. This can simplify instrument writing, or at least allow one to set up prototype instruments quickly that can then be later modified. One may simply type in parameters into labeled fields of the StochGran interface and press buttons to run the job and play the Cmix sound file. The control of thirty-six parameters is made more convenient to view. To explore the implications for the music when an appropriate graphical user interface is used, the compositional process used to create "Dragon of the Nebula", a tape composition, will be described. This is, of course, not a statistical analysis of the usefulness of compositional interfaces, but rather one composer's experience.

Patchmix

Patchmix was designed to bypass writing C code while constructing instruments within the flexible

ICMC 563
Cmix programming language, developed by Paul Lansky at Princeton University. For people learning C, or those who would not like to spend the afternoon (at least) debugging a program, Patchmix will allow one to set up complex instruments quickly.

The user selects unit generators from a menu, and places the icons onto a window. The unit generators are then connected, outputs to inputs, on the graph. When finished, the flow chart can be compiled and tested. The Cmix code is written out into files with a user-specified directory. After the new instrument is compiled, the user can specify score values for the instrument, save and test and hear the sounds. Since the
Cmix code is written out, it may be modified later and recompiled outside of Patchmix. For someone learning C and Cmix, s/he can study the instrument and learn how to write other instruments in C programs.

Patchmix has so far been useful in these situations: 1. To set up a prototype for a more complicated instrument quickly, with the composer adding modifications or additions later. 2. To put together instruments with a number of unit generators, which otherwise would take time writing many lines of C code. 3. To demonstrate simple instruments, so that a student can see how to use Cmix.

This program was originally done in C++ under X-Windows, and now is available in NoXtStep.

**StochGran**

StochGran was written to create granular synthesis sounds with "stochastic" parameters. For a description of granular synthesis techniques, see Curtis Roads [1988] and Barry Truax's [1988] articles. "Grains", or very short sounds with an envelope, are synthesized that have rates, durations, locations and frequencies which lie within certain specified ranges during an event. These parameters may change during the event, and may vary by a certain amount. The user can specify the mid point and the low and high boundaries of the parameters chosen. Also a "tightness" around the midpoint is chosen. A simple probability function will generate values within these limits which hug the midpoint with the desired degree of tightness. While not sophisticated in terms of statistical methods, these parameters of control over the event give the composer quite a bit of flexibility in timbral generation. This program was first done as a Cmix instrument called "igran". Then, a StochGran NoXtStep interface was built to the instrument which wrote out Cmix score files, ran the jobs and played resulting sounds. It presents the user with a window of labelled text fields to input values, and menus from which to select the commands to run the Cmix process and play the resulting sounds. A newer version will use NoXt music kit objects to synthesize sounds in real time. A future project is to incorporate live sampling and granular processing on the incoming sounds and "in-phase correlation" of the grains.

Another addition to StochGran will be the ability to draw a shape with a mouse, and have the program translate this line into a function for the parametric change shape. For example, if the average frequency of each grain's waveform is to rise in an uneven line with several dips, one will be able to draw the exact shape with the mouse. Currently one must calculate waveable locations or compromise and use the "Flat", "Linear" or "Exponential" functions built in to the interface. While the composer may continually have new ideas and need to do more programming, today's GUIs simplify this task.

**Dragon composition**

StochGran was used to create much of the composition, "Dragon of the Nebula". In a previous piece, "Song for Earth Day", I had used the Cmix instrument "igran", editing score files with thirty six floating point values for each event. While shaping the sounds, I would accumulate many related versions of sounds with similar names and just one or two parameters changed. Keeping track of all these sounds was
difficult and tended to limit the complexity of the piece. But for the new piece "Dragon", after getting a new idea, I could test it out and hear it in a few seconds. I tended more often to keep just the longer more shaped sequences since I could so quickly regenerate test sounds with StochGran. It was easier to find timbral relationships between different sounds by "tweaking" the parameters within the interface. Because of these factors, "Dragon" explored more timbral territory and I was able to spend more time processing the sounds with reverberation and filtering, since less time taken to generate and organize the sounds. Other ideas for additions to the program would arise during the work on the piece. By the end of the piece the composing involved a new C program to generate multiple score files. I am currently adding this algorithmic score generator into a window of StochGran. The program "dpf", for "dragon phrase" will produce scores with their low, mid, high and tightness values falling within certain allowable ranges. "Dph" uses the probability function one level higher to determine the ranges for the ranges to be written out into the score file parameters. In this way I could, alter much adjusting of the range parameters, algorithmically generate fifty or more scores, many with completely different characteristics. Because some of the grain rates could become quite dense, and paired with long durations could require hours to generate, I saved files that were calculated to be "very long" until later. Listening to the shorter files, I was amazed at the unexpected gestures that were created. It was one of the more exciting experiences composing that I have had, and fit well with the idea I had of a powerful, evolving sequence of gestures for this piece. I was able to take many of these sounds and meld them into new, complex sounds of which I might not have conceived, had I been editing score files. It would have taken weeks rather than days to put together the final section of the piece without these programs.

Conclusion

These two interfaces can provide environments which are convenient for composition. While the interface may not contribute too much to the musical ideas being expressed, it should at least not hinder their realization, as has the text interface in the past. These two programs were designed to allow the composer to focus as much as possible on the ideas, and let repetitive tasks such as writing syntactically correct code and typing lines of note data be done by the program. Patchmix has been used to teach Cmix programming, and to set up prototype instruments which can be altered later for putting together sounds. StochGran has made possible a more complex depth of timbral work, in the composition of the piece "Dragon of the Nebula". The new ideas generated by these programs continue to suggest expansions and modifications. By spending the time to create tools specific for a piece or set of pieces, one may actually make the process of composition more intuitive and successful.

Both programs may be obtained by sending a request to mara@woof.columbia.edu.

References


Default StochGran window, with comment added.

A sample Patchmix window showing connected envelope generators and oscillators with frequency modulation.