

# *The Musicians' Software Mall*

## A Set of Composition and Performance Oriented Applications for Sound Synthesis

Eduard Resina, Xavier Serra  
Audiovisual Institute, Pompeu Fabra University  
Rambla 31, 08002 Barcelona, Spain  
{eduard, xserra}@iaa.upf.es http://www.iaa.upf.es

### Abstract

The Musicians' Software Mall is a software package developed to provide powerful composition and performance tools to the computer musician. At this stage it contains the following applications: SmsTools 2.0, a set of tools for Spectral Modeling Synthesis analysis of sound files; SmsPerformer 2.0, a graphical interface for the real-time SMS synthesis engine; Music Maker 2.0, an application for composing, generating, editing and synthesizing either SMS or CSound score files; Drizzle 2.0, a graphical interface for controlling real-time granular synthesis. In this article we concentrate in two of these applications: SmsPerformer and Music Maker.

## 1 Introduction

Spectral Modeling Synthesis (SMS) [1], provides musically attractive and powerful tools for sound transformation. The SMS synthesis is based on the combination of additive and subtractive synthesis.

With the power of current general purpose processors it has become possible, and highly appealing, to apply this system to new and more complex musical situations than it could be attempted some time ago. Since 1998, some applications designed to control and manipulate the synthesis parameters of SMS analysis files have been developed at the Audiovisual Institute of the Pompeu Fabra University of Barcelona. SmsTools, a Visual C++ code with graphical interface implemented by J. Bonada [2], SmsPerformer [3], and Music Maker (formerly SmsComposer) [4], form a trilogy of such SMS related applications, which are a continuation of the software implementation of SMS started by a C code program with a command line interface developed by X. Serra. Drizzle (formerly Vocem) [5] is a graphical interface for controlling the parameters of real-time granular synthesis.

Where as SMSPerformer has been designed for real-time performance, Music Maker stands by a different and somehow opposed kind of musical premises and demands. Music Maker is a score (list of events or parameter specifications for the synthesis of some SMS analysis file) generator, editor and synthesizer. It is not designed for real-time synthesis, but rather for detailed editing of the events' synthesis parameters. These two applications are explained in more detail next.

## 2 SmsPerformer

SmsPerformer can run the synthesis and all available transformations in parallel and in real-time in a Pentium Processor 200 MHz, 32Mb RAM. It has many ways to control the synthesis-transformation of the sound: the sliders' window, a score file, a MIDI file, a MIDI controller, a set of graphical envelopes, or a predefined set of algorithms.

### 2.1 The sliders' window

The most immediate way to use the program is scrolling a number of sliders. You can configure which synthesis-transformation parameter controls each slider and set the maximum and minimum values of the slider's parameter in the sliders' window. (Figure 1).

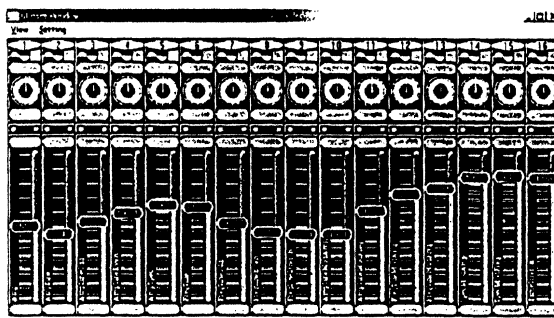


Figure 1: Sliders' view.

The slider values can be modulated by a sinusoidal signal or by a uniform random noise.

## 2.2 External MIDI controls

The application accepts MIDI messages for changing the transformation parameters from an external MIDI controller. In this way you can get a better physical feeling (like playing an instrument) than using the main window sliders. It is also possible to change more than one parameter at a time.

A MIDI keyboard can also be used to set any of the various frequency synthesis parameters. Key velocity and aftertouch can be mapped to control other parameters. Note-off messages can be set to interrupt synthesis. In this case, a note-on message will restart synthesis from where it was left.

## 2.3 MIDI and Score files

You can save performances as MIDI files and play already saved performances by loading them. The MIDI file saves the value, the slider (channel) and the time of a value change. When playing a MIDI file you can act on it modifying some parameter values at the time you hear the saved performance.

SmsPerformer can also read sms score files: synthesis files (\*.syn) and hybridization files (\*.hyb). These files are text files in which you can specify all sms synthesis and transformation parameters [6].

## 2.4 Setting sliders' configurations

You can also modify the sliders' value by drawing with the mouse the configuration you want the sliders to have. Dragging the mouse on the control (shown in *Figure 2*), once the control has been activated can do this. Each bar in the control can be independently activated or deactivated by clicking on the number at the bottom of the bar. This filters input while drawing on the control.

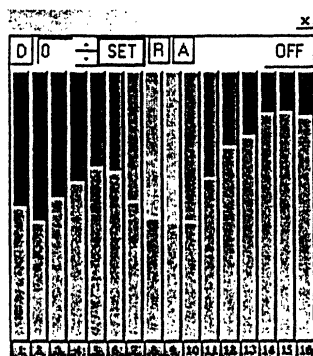


Figure 2: Sliders' configuration window.

Each configuration can be saved and referred back as a preset in a slider setting envelope.

## 2.5 Envelopes

You can organize a performance by defining an envelope. This envelope is defined in a x-time by placing any defined setting (represented by a box with the setting number) on the time-rail (*Figure 3*).

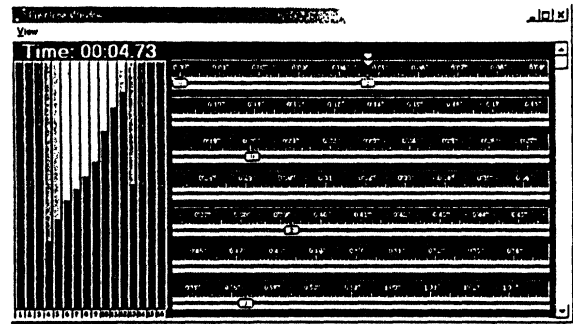


Figure 3: Envelope Window.

The values of the sliders can be either interpolated from one setting to the next or hold up to the following setting. Both time and preset values of an envelope can be modified while the sound is playing.

Another kind of envelope is the one provided by the Track Window (*Figure 4*). The user can define any number of tracks or envelopes to precisely draw the trajectory of any synthesis parameter during performance. The tracks can be muted, played solo, and drawn when edit mode is selected. Each track has 128 values in the y-axis (which correspond to MIDI values), and the output parameter value is mapped according to each parameter's range. This allows to copy/paste from one track to another since the proportional values can be applied to any parameter.

These envelopes provide SmsPerformer with a compositional environment that provides detailed and pre-definable control of a performance session.

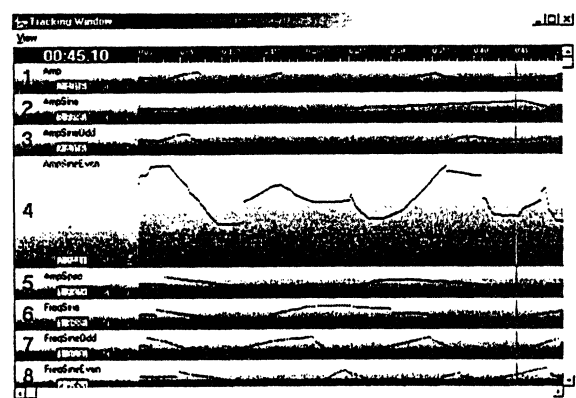


Figure 4: Tracking window.

## 2.6 Macros

SmsPerformer includes a macro defining window (*Figure 5*) with a series of objects to map parameter, slider, midi input/output, and algorithmic values. This allows, for instance, to control several

parameters with one single slider. Values from one performance can be output via MIDI to control another performance run in a different computer.

There are C language type of objects (switch object, if/else-if/else object) and a general interpreter object that allows the user to enter C language code to define the behavior of a parameter.

The defined macros can be assigned to a slider just as a parameter can, and the input to the macro be provided in any of the ways described above.

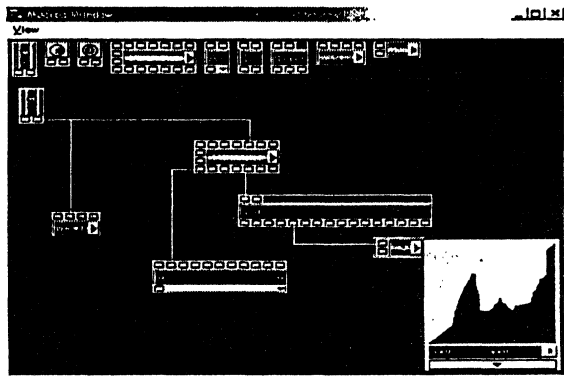


Figure 5: Macros window.

### 3 Music Maker

The design of Music Maker responds both to demands derived from previous compositional experience using SMS [7][8][9], and to the influence of some other score generation environments used through these last few years (especially J. Rahn's Lisp Kernel [10] and R. Taube's Common Music [11]) for CSound score generation, as well as the Lisp embedded list philosophy which turns to fit so nicely into the proliferation of musical levels so natural to compositional thinking.

#### 3.1 From the beginning

When the user starts a Music Maker new session, an ordered list or map of all the available synthesis parameters appears on screen. When the application writes the score file, only the selected (highlighted) parameters will be taken into account. (Figure 6)

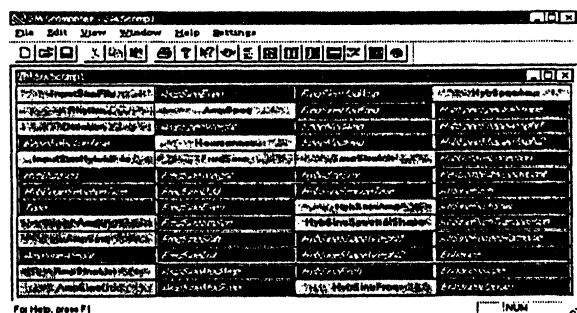


Figure 6: Main window.

Before any of the parameters can be edited, the user will be prompted to enter the path of at least one SMS file. In order to edit any of the hybridization parameters the user must enter at least another SMS file for this purpose.

Both InputSmsFile and InputHybridFile dialog boxes can register up to twenty different SMS files. The user will be able to assign any of these synthesis and hybridization files to each individual event or group of events to be generated in the score.

#### 3.2 The Envelopes

By double clicking on any of the parameters on screen, an envelope window with a double (eventline vs. timeline) mode is shown. Most of SMS parameters can take either a single value or an envelope as input. The window for these parameters can be switched to envelope type of input.

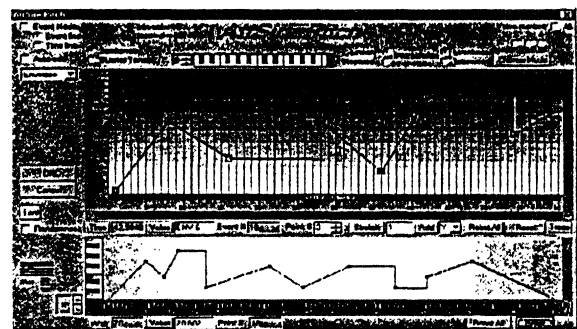


Figure 7: Control window for one parameter.

In such a case, the points on the main envelope refer to one of the twenty secondary envelopes the user can define per parameter. Every parameter's main envelope can edit up to four independent voices or sequences of events. Every voice has an event or time offset, as well as an independent amount of randomness, which will be applied to the voice's values if selected. (Figure 7)

The points can be entered directly into the envelope through mouse clicking, by playing on a MIDI keyboard, or (in the case of the various frequency and duration parameters) by mouse clicking on a graphical keyboard representation, or by selecting musical notation duration values which can represent a large range of rhythmic combinations including triplets, quintuplets and septuplets. When using musical notation for the duration (instead of absolute decimal notation), the user will be able to change the duration ratio by modifying the global tempo at will. Events between points can either interpolate or hold the last point's value, and a number of editing tools to set and modify the value of a group of points is provided.

### 3.3 Lists and sublists

The recursive nature of Music Maker is implemented in such a way that every envelope point object owns a pointer to a list. This is, every point can become a list, and every List object can hold any number of points, which in turn can become sublists. This way the user can descend down a tree of several levels of sublists.

Music Maker inherits some of the item stream types implemented in Common Music (cycle, sequence, rotation, palindrome, random, etc.) The user can keep track of the ramifications and visualize a representation of the sublist tree. The graph will show the type of list, its period, as well as all its descendants. (Figure 8)

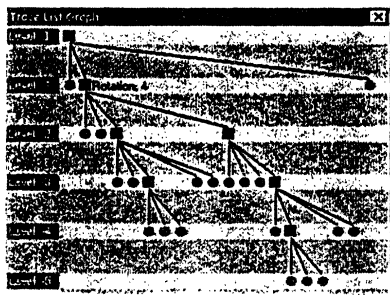


Figure 8: Trace List Graph.

### 3.4 Ending the session

Once the score has been generated, it can be edited from within the application. The editor can search any voice's event either by event number or by event time, making editing of the score fast and easy. Finally, the user can synthesize the score and listen to the resulting sound from within the application.

Even though Music Maker was originally designed to suite the demands of composing with SMS, the structure of the application made it easy to adapt it in order to generate CSound scores. Music Maker 2.0 has incorporated a number of tools for generating and editing CSound scores and orchestra files, as well as defining GEN tables, envelopes and instruments.

## 4 Conclusions

SMS has, for some time now, been developed and researched with few applications to composition and performance. The musical practice world imprints a series of demands that are currently being translated into user applications, which will allow musicians to take advantage of the great potential that SMS offers them in their various creative tasks. These applications are intended to enrich the possibilities of both computer generated works and computer assisted performance.

## 5 Acknowledgments

We thank all the research team of the Music Technology Group at the Audiovisual Institute of the Pompeu Fabra University in Barcelona, for their contributions, help and support.

## References

- [1] X. Serra. "Musical Sound Modeling with Sinusoids plus Noise". G. D. Poli and others (eds.), *Musical Signal Processing*, Swets & Zeitlinger Publishers, 1997.
- [2] Music Technology Group. *SMS Web site*. URL: <http://www.iaa.upf.es/~sms>.
- [3] E. Resina. "SMS Composer and SMS Conductor: Applications for Spectral Modeling Synthesis Composition and Performance". *Proceedings of the Digital Audio Effects Workshop (DAFX98)*, 1998.
- [4] A. Loscos and E. Resina. "SmsPerformer: A Real-Time Synthesis Interface for SMS". *Proceedings of the Digital Audio Effects Workshop (DAFX98)*, 1998.
- [5] Daniel López, Francesc Martí, Eduard Resina. "Vocem: An Application for Real-Time Granular Synthesis". *Proceedings of the Digital Audio Effects Workshop (DAFX98)*, 1998.
- [6] X. Serra and J. Bonada. "Sound Transformations based on the SMS High Level Attributes". *Proceedings of the Digital Audio Effects Workshop*. 1998.
- [7] E. Resina. *L'Esquizofrènia dels Sons*, for reciter, septet, and computer generated sound, 1993-94. (SMS running on NeXT).
- [8] E. Resina. *si eSe Me vieSe*, 1996. (SMS running on NeXT).
- [9] E. Resina. *Menstruació*, for female voice and computer transformed voice, 1998. (SMS on Windows).
- [10] J. Rahn. "The Lisp Kernel: "A Portable Software Environment for Composition", *Computer Music Journal* 14(4):42-58, 1990.
- [11] H. Taube. "Common Music: A Music Composition Language in Common Lisp and CLOS", *Computer Music Journal* 15(2):21-32, 1991.