IRIN: Micromontage in Graphical Sound Editing and Mixing Tool

Carlos Caires
CICM – Centre de Recherche Informatique et Création Musicale
Université de Paris VIII
carlos.caires@wanadoo.fr
2, Rue de la Liberté 93526 Saint Denis Cedex 02, France

Abstract

Micromontage technique allows the composer to work a musical figure point by point, shaping each sound particle with microscopic precision. The software presented in this paper combines graphic and script editing with algorithmic generation and manipulation of sound sequences. It provides several tools to enhance both creation and organic development of the musical material under this compositional paradigm using a user-friendy visual environment

1 Introduction

IRIN is a composition tool implemented as a Max/MSP standalone (Zicarelli 1998), designed to enhance several composition operations enclosed in the *micromontage* paradigm[1]. It uses a comprehensive user interface containing several editing windows allowing the composer to assemble, view, playback and modify several kinds of sound objects.

Generally speaking, micromontage technique consists of the extraction of sound samples from sound files and then rearranging them in time. Each extracted sound sample can be multiplied and transformed through operations like speed variation, filtering, panning and amplitude envelope editing (Roads 2001). These very simple sound processing operations are typically enough to build a wide-ranging sound catalogue. Chosen groups of samples from the pre-composed catalogue are then arranged in time into more complex sound structures on a higher time scale that can subsequently be also transformed and multiplied through several variation operations (Vaggione 1995; Vaggione 1996). This kind of compositional approach calls for a working environment that is able to keep track of all important sound operations. Taking into consideration that sound transformation, far from a mere "effect", is an act of composition, a memory of all actions and respective data involved within the process of creating the tiniest sound particle is needed so that a consistent proliferation of the musical material can be achieved. All data concerning sound manipulation is therefore stored and accessed through IRIN's graphic interface. IRIN was designed to offer control over micro-time and macro-time levels of composition, allowing the composer to browse smoothly between them with an analogous outlook towards musical material.

2 Basic IRIN features and use

The functioning of IRIN can be summarized as follows:

1. Load up to 16 Sound files of any size depending on the available RAM.

2. From each sound Buffer, select a region and edit it in several ways to obtain a Sample.

3. Place samples on any one of the 4 available tracks, knowing that tracks are polyphonic and sound parameters are a track independent feature.

4. Store edited Samples in a sample library. Every stored Sample (in a track or in the library can be retrieved for later use or further manipulation).

5. Encapsulate sound sequences into a Figure object and submit them to several variation operations. Store Figures in a library.

6. Encapsulate up to 8 Figures into a Meso-structure object and submit it to several variation operations. Store the Meso-structures in a library.

7. Add customizable shapes and colours to every sample. Shapes are used for display in the Timeline window on "shapes view" mode.

[1] IRIN was presented for the first time as "Mixage" (Caires 2003). Besides introducing several new features, this version now runs under MacOs X.

8. Control all operations graphically and use a Timeline to place sound events in time.

9. Render the final result into a multi track audio file.

3 Classes

Concepts like cell, figure, counterpoint, polyphony, layers, background/foreground, variation/repetition, and so on, are permanently involved in the process of composing, as much in instrumental music as in electroacoustic music. IRIN helps one to compose and organize sound material having these kinds of concept in mind. Sound objects were organized hierarchically into three different object classes: Sample, Figure, and Meso-structure. Each class inherits the properties of its predecessor adding new ones belonging to the domain of a higher time scale. Together, Sample, Figure and Meso-structure objects form the pre-composition materials yet to be organized into the macro-scale form of the piece. At the top level, there is a fourth layer named Timeline. The Timeline represents the “final” score where all objects (from the sample to the Meso-structure) are arranged in time to be played back or exported as an audio file.

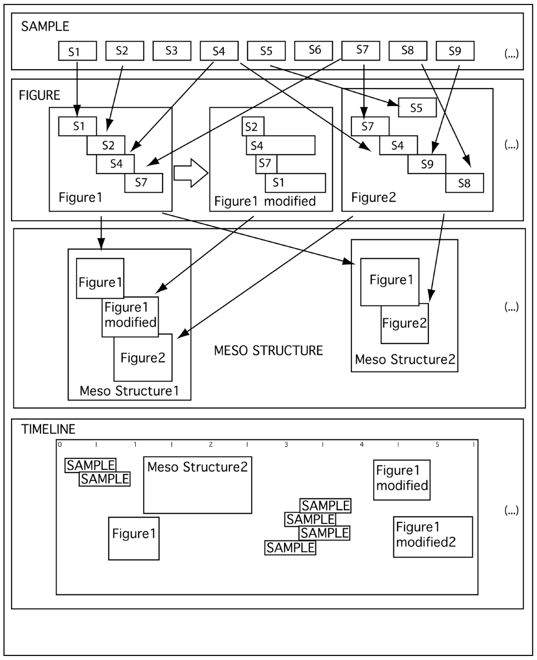


Figure 1: Tree-like representation of sound object dependencies

3.1 Sample

The Sample is the simplest of IRIN’s sound objects. This atomic object contains six properties editable from the Sample edit window (Figure 2):

Source sound file. The name of the file along with the sample start and end points.

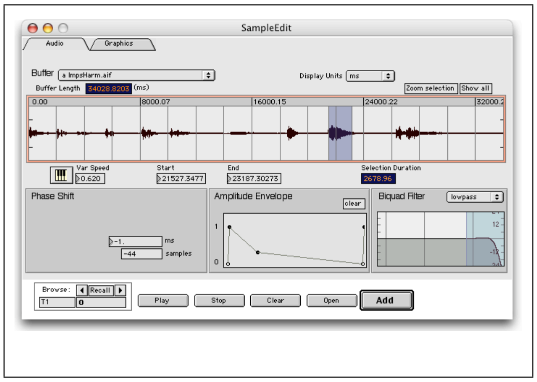
Speed variation. Time-domain speed variation: changing speed also changes pitch.

Filter. Implemented using Max/MSP *biquad* object (digital filter specified by coefficients).

Amplitude envelope. Controlled by a breakpoint function editor accepting up to 256 points.

Phase shift or multi channel trajectory. Depending on the current audio output configuration one can choose to assign a phase shift value [2], or edit a multi channel trajectory.

Shape. Each sample can be assigned to a preset graphic shape of any color (shapes are used for display in the timeline window on “shapes view” mode).



Buffer: a ImpsHarm.aif
Display Units: ms
Buffer Length: 34028.8203 (ms)
Var Speed: 0.620
Start: 21527.3477
End: 23187.30273
Selection Duration: 2678.96
Phase Shift: -1. ms, -44 samples
Amplitude Envelope
Biquad Filter: lowpass
Browse: T1 0

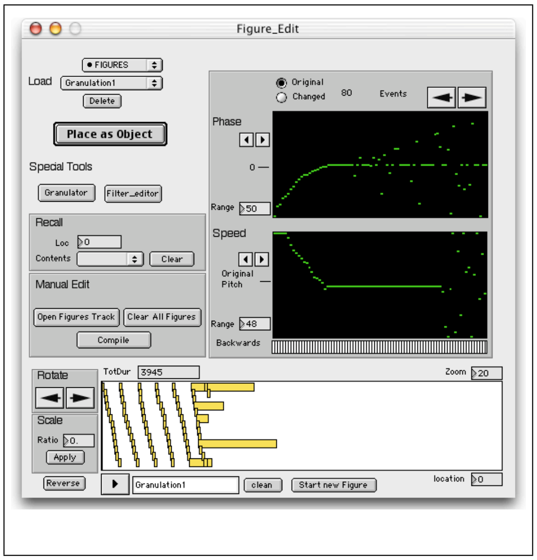
Figure 2: Sample editor window.

3.2 Figure

An array of Samples in a specific order, each one anchored to a specific onset time can be encapsulated in a Figure. Inside a Figure, the order of the events and their onsets

[2] Phase shifting in this context is used as a composition technique belonging to the micro-scale domain (*micro decorrélation temporelle*). It concerns the definition of spatial attributes of small sound particles by placing micro delayed replica of it into different channels (Vaggione 2002b).

are changeable. This means that each point of discontinuity is a target for a compositional action (Schilingi 1998). Because this class inherits the properties of the Sample class, each one of the Sample properties described in the previous section is also modifiable inside a Figure. Its edit window (Figure 3) comprises several tools allowing a quick assignment of new values for phase shift, speed variation and filtering to the entire figure. Among them we can find two resizable multi-slider panels for speed variation and phase shift edit, a break-point filter editor to filter evolution within the Figure, several reordering, scaling and reverse tools that can affect the whole figure or just one of its attributes.



Load: Granulation1
● Original ○ Changed 80 Events
Phase Range: 50
Speed Range: 48
Recall Loc: 0
Ratio: 0.
TotDur: 3945
Zoom: 20
location: 0
Granulation1

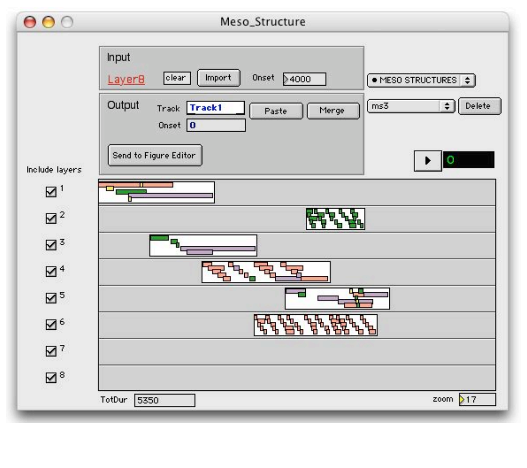
Figure 3: Figure Editor window.

A granulator is also available inside the Figure Editor. It can be used to output a stream of a pre-defined number of particles with a particular behaviour as regards the evolution of their properties. The evolution of the duration of its Samples, of the distance between them, of their phase shift and of their filtering can be set before outputting into a Figure object. Despite the use of the *global laws* strategy, the Figure obtained can be then locally changed “by hand” since each one of its elements is an instantiation of the Sample object that, as previously mentioned, can be modified in several ways. Direct manipulation, therefore, plays an essential role in this context, since morphological properties of a given Figure (*gestalt*) arise from its local singularities (Vaggione 1996; Vaggione 2002a). Eventually, after all local changes have been performed, a Figure results as a configuration (or pattern) so unified as a whole that it cannot be described merely as a sum of its parts.

How to achieve this is already a compositional issue.

3.3 Meso-structure

Meso-structure is the locus for polyphonic development of more complex Figures (Figure 4). Just like the interdependency that has been established between Samples and Figures, a Meso-structure can encapsulate several Figures. Its editing window comprises an eight-layer sequencer where individual onsets can be assigned to each contained Figure. All objects inside a Meso-structure keep their individual methods and properties. A Figure remains a Figure and can therefore be modified through its own editor which, in turn, gives access to the Sample editor. In fact, the three main IRIN editors (Sample editor, Figure editor and Meso-structure editor) are permanently linked. The way IRIN’s interface was built directly reflects all class dependencies shown in figure 1.



Input: Layer8 Onset 4000
Output: Track Track1 Onset 0
ms3
0
TotDur: 5350
zoom: 17

Figure 4: Meso Structure Editor window.

4 Timeline

At the top-level we have the Timeline window. Timeline is basically a 4 track sequencer were Samples are represented as a coloured bar, their size reflecting their duration (Figure 5). Figures are displayed as a white rectangle labelled with the respective figure library name.

The notion of track used is somehow different from the one we often find in traditional audio multi-track sequencers. For instance, whether in ProTools or in Digital performer, panning and volume curves are edited in the track affecting the sound blocks placed in background. In the case of IRIN’s timeline, because each sound object keeps track of its internal data, their properties (including panning and envelope amplitude) are a track-independent feature. Thus, every event always keeps the exactly same properties if moved back and forth inside a track or if it is changed between tracks.

Moreover, tracks are polyphonic. The main advantage in this approach is that all kinds of object can be superimposed in complex configurations and yet all events remain visually close to one another. Figure 5 shows an example of this: in track one, for instance, two figures appear over a set of samples. Local sound agglomerations are easily tracked at a quick glance, since the materials can be arranged as if they were in a score, overcoming the usual multi-track distribution constraints. As a metaphor for score staffs, tracks may be used precisely as a compositional tool, helping the composer to arrange the polyphonic stratification of his material in a more systematic way.

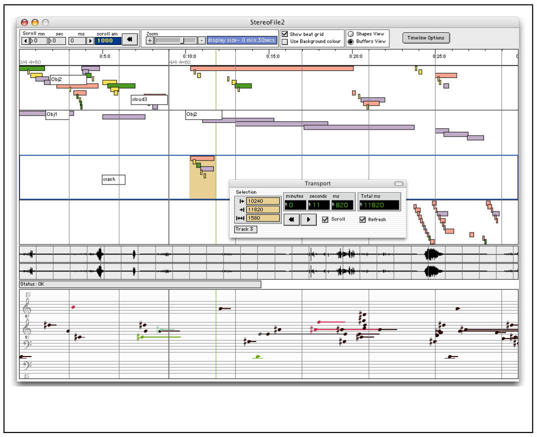


Figure 5: Timeline window.

Finally, Timeline includes two extra tracks: a Sound file track and a MIDI track. Once included in the Timeline window, they are played along with the contents of the upper tracks. MIDI track provides a limited representation of score notation where all midi channels, represented by different colours, become merged into a four staff score.

5 Discussion

The set of available object transforms is being extended to include more advanced operations than those included in this version. It is important to mention that since all data concerning Sample, Figure, Meso-structure or track, is stored inside Max *coll* objects with specific labels, it is relatively easy for experienced MAX users to develop custom functions, making them available to IRIN. For whoever may be interested, it is also possible to implement global post audio treatment to each track, simply by intercepting the audio running inside IRIN. This can be achieved using Max *send* and *receive* objects.

Detailed documentation on these subjects is being prepared.

6 Acknowledgments

I should like to thank my supervisor, Horacio Vaggione, for the advice and continued support. I would like also to thank Christopher Bochmann for reviewing the draft of this article.

This research is supported by “FCT e FSE no âmbito do III Quadro Comunitário de Apoio”.

References

Caires, C. (2003). Vers une écriture des processus de micromontage. In *Jornées d’Informatique Musicale*, Montbéliard.

Roads, C. (2001). *Microsound*. Cambridge, Mass. ; London: MIT Press.

Schilingi, J. B. (1998). Morphologie et structures musicales. In Documentation (Ed.), *Fonctions d’analyse, de reconnaissance, de classification et de reconstitution de séquences symboliques et numériques*. Paris: IRCAM.

Vaggione, H. (1995). Objets, représentations, opérations. *Ars Sonora 2*, 33–51.

Vaggione, H. (1996). Vers une approche transformationnelle en cao. In C.-U. d. C. Les cahiers du GREYC (Ed.), *Journées d’Informatique Musicale 1996*.

Vaggione, H. (2002a). Composition musicale et moyens informatiques: questions d’approche. In *Formel Informel*, pp. 99–117. Paris: L’Harmattan.

Vaggione, H. (2002b). Decorrélation microtemporelle, morphologies et figures spatiales. In ADERIM-GMEM (Ed.), *Journées d’Informatique Musicale 2002*, Marseille.

Zicarelli, D. (1998). An extensible real-time signal processing environement for max. In *International Computer Music Conference*. University of Michigan - Ann Arbor.