Sound and musical representation: 
the Acousmographe software

Yann Geslin, Adrien Lefevre
INA - Groupe de Recherches Musicales, Paris
ygeslin@ina.fr - support@adlef.com

Abstract
The “Acousmographe” software has been developed by the GRM’s research team since 1991 to allow graphical representations of sound. This software is specifically useful for annotating, analyzing and representing non-traditional music, in particular music that doesn’t lend itself to traditional score representation. This software has been widely used by musicologists, teachers and web designers to build interactive representations of electroacoustic, ethnic, jazz or improvised music. A fully revised version will soon be released, that facilitates access for non-expert users while including numerous new professional features. After a general and historical presentation, the new data structure organization will be described and discussed.

1 Sound transcription
Occidental music is generally represented by a score that indicates the operations necessary to perform a piece. This score is also used by musicologists to analyze the musical content in terms of structural properties. In the case of non-written music (ethnic, electroacoustic or improvised), the description and analysis of the musical content is not easy. A graphical medium is essential and has to be realized: this representation is not a score, but a sound transcription (Roads, 1995). This is in the sense that the result is a representation of what one hears (nobody will try to play the music from reading its representation), and that the visual aspect of the representation is fully dependent on the transcriber’s choices and points of view (Besson 1991).

Furthermore, if one wants to analyze and compare several interpretations of a classical piece of music (i.e. a Bach prelude by famous pianists), the score is available but does not help much in characterizing the sonic differences that are perceived by the listeners (Delalande 2001).

Over the years, experiments have revealed certain directions in the way one can transcribe sound and music recordings: first possibility is a coded representation of the elements, and the second is a spectro-morphological representation of sonic parts. The coded representation uses a set of elementary and abstracted units related to the sounds; it needs a reference sheet that explains the arbitrary chosen relations between the drawing units and the sounds; some of the bests examples have been recently developed by Lasse Thoresen (Thoresen, 2002, Couprie, 2003) (see figure 2). The spectro-morphological representation attempts to draw and highlight some of the dynamic and spectral figures of the sounds (Smalley, 1986, Bayle, 1993). It usually needs an FFT spectral background representation that improves our perception and comprehension of the spectral nature of the sounds. But this spectral approach misses lots of perceived information such as transients.

Figure 2. Lasse Thoresen coded transcription of Ake Parmerud “les Objets Obscurs”

Another main issue addresses the questions of the time to sound relation: does the sound transcription have to be a dynamic or static representation of the time? A static representation is easier to realize, but of course misses some obviousness of sound to spectral lack of accuracy. On the opposite, the experience of adding a time cursor to some achieved representation (i.e. the Ligeti’s Artikulation transcription by Rainer Webinger or the Stockhausen’s Studie II genuine score) shows that one can accept or reject some representation because of the lack or confusing excess of synchronized representation.
2 The Acousmographe history

Since the fifties, the GRM’s composers and researchers have made a frequent use of sound representation, either for the study and description of sounds properties (in case of the need of a music representation at concerts) or even as a necessity to deposit their music at the Musical Rights Society. The first sound representations were made manually or with the help of early sonograms capacities. In 1972, an experience of an animated and graphical representation was realized by François Delalande and Jacques Vidal (Delalande, 1972), showing the interest of a time-synchronized representation. At the end of the eighties, the need of a specific tool for sound and music representation was obvious, and the help of the early computer drawing software was demonstrated, in particular by Denis Dufour on his MSX™ personal microcomputer for the analysis and teaching of the electroacoustic repertoire.

The first realization of the so-called Acousmographe software was made by Olivier Koechlin around 1990 on a request of the GRM’s director and composer François Bayle. It was a prototype for the Supercard™ Macintosh graphical environment that could superpose an FFT representation to vectorial white and black figures on a 5 to 10 minutes sound file. Simultaneously to the graphical representation, the sound file could be listened on a Dyaxis™ card and later on the first Digidesign™ Sound Designer audio card (Koechlin and Vinet 1991). The software was intensively used despite its instability and limited experimental capacities; it was also the base of an interactive CD on sound representation and animation (Besson, Koechlin and al 1993).

3 The need of new features

In 2003, the GRM was funded by the French Educational Ministry to provide a new version of the software that would be used in all schools for musical and non-musical purposes. On the other hand, the experiment of the two former releases of the software by very different and professional users gave rise to a huge amount of expectation and requests for a new version. Some of those requests seemed not compatible, as for example the need for more power and professional features and on the other hand for more simplicity. A fully new development of the Acousmographe was decided and then started in December 2003 by Adrien Lefevre for the main programming aspects, under the direction of Yann Geslin and Emmanuel Favreau. The new release is expected for December 2004. The team decided to split the features in two frames: one to offer a better access to inexperienced users, and a second to better connect the software to common professional tools like score writing, drawing and multimedia software (Vinet, 1999). The core of the software will still be independent and integrate the most useful and basic tools needed to make or show a full sound representation. Then a plugin structure...
will allow to import/export the most necessary parts of the representation from the Acousmographe to other software.

However, some very specific aspects remain, that have to be solved in the core. The first is the use of very long sound files and FFT files, with multichannel capacities and very good time-synchronization. A second is the ability to realize and lock a sound representation to an identified commercial audio CD; this is for author’s rights protection reasons as well as the necessity of reducing the required size of the files to download. The third is to allow very protected uses, for example by children and non-expert users, and to offer a slide-show capacity that can be used by the teacher during lectures. And the last one is to offer a better integrated and automatic sound segmentation, so that a draft sound representation can be obtained without too much time or knowledge.

For the expert uses, it was observed that a significant part of the users employed the Acousmographe as a starting point, doing long FFT and automatic or manual segmentation, then manually exporting the information under their favorite graphical and multimedia software (see figure 5). Besides this, the decision was made that it was impossible to a very small team to offer lots of the professional tools that experts want to freely use. As far as possible the connection will be kept between such heterogeneous software, by allowing the exportation of the FFT representation layer, timeline data, segmentation detection and/or time and position lists of any drawing layer.

![Portraits polychromes](image)

**Bernard Parmegiani**

**Mouvement 5 “Signes”**

![Figure 5. Acousmographe + GraphicConverter transcription of Parmegiani “L’Oeil Écoute” by Philippe Mion (2004)](image)

#### 4 Software architecture and implementation

##### 4.1 Specific problems and objectives of development

The development of the new Acousmographe (version III) is directed around several major stakes.

First, the Acousmographe is fully a software of graphic symbols edition; the diversity of the symbols for the representation of a timbre or a sound is potentially infinite. These symbols can be of several natures (bitmaps, vectorial 2D, even 3D), under very different data formats (Illustrator, Finale, GuidoLib, etc). This requires an open architecture, allowing the addition and importation of new types of symbols, without having to modify the data-processing code of the Acousmographe; it constitutes typically an architecture with plugins.

Secondly, the Acousmographe files (transcriptions, libraries) can contain a considerable number of graphic symbols, and the sound files (and thus spectrograms) can have durations of one or two hours. The size of the data to be treated is thus substantial (about 100MBytes in some GBytes); the traditional solution consisting of loading integrally the file in the memory when opening a document is not suitable, and specific solutions must thus be considered.

Lastly, the software must run under Windows XP, but also if possible under Mac OS X. The user interface of the Acousmographe must be standard, reactive, intuitive and simple to use. In particular, powerful undo/redo commands are implemented, and a history feature of user actions should be achieved.

##### 4.2 Solutions

The Acousmographe is divided into two distinct software parts: the core and the graphic user interface. The core contains all programs of pure processing data and the
user interface is the software part allowing the control of the core. Moreover, the core treats the operations in the form of standardized "actions"; in return it is able to send messages to the user interface for some graphic refresh stage (see figure 6).

Aiming a maximum portability and ensuring the recognition of the data-processing code by the greatest number of compilers, the core is written in a C-ANSI language. This guarantees good performances and allows the implementation on WindowsXP, MacOSX, and possibly on other Unix.

Several solutions are considered for the realization of the GUI (Graphic User Interface) of the Acousmographe, by taking into account that it must be developed in parallel on MacOSX and WindowsXP. Moreover, the plugins also comprise modules of GUI (preferences panel, inspector, etc). The choice of an adapted cross-platform library of GUI (wxWindows, PowerPlant, FoxToolBar, etc.) is thus crucial for the effectiveness and the continuation of the software and to preserve the simplicity of development of the plugins.

Figure 6. Main modules of the Acousmographe

4.3 Plugins and objects

The choice of a plugins architecture in the Acousmographe is essential, because it is the only valid mean that ensures in the future importation capacities of new types of graphic symbols. It acts as an open software, comprising at its base only one central control device - a container - whose content is managed by plugins. The plugins behave as many software elements that come and connect themselves to this central control device; they can be developed by other teams, each one bringing a new symbolic notations to the software.

The manufacture of the plugins must absolutely remain simple, if not, the option to open the software would become immediately obsolete. For this purpose, and parallel to the development of the core, we carry out an API (Application Programmer Interface) that is simple, clear and documented.

The plugins of the Acousmographe line up in two principal categories: graphicals and calculators. The first allow the representation of the graphical symbols, and their management in the form of objects of class. The seconds are used to carry out calculations in a specific way: import of graphic symbols under various formats, export of the transcriptions (scores) in the form of films, calculations of spectrogram FFT or wavelets, automatic positioning of temporal markers, etc. Even if the plugin API of the Acousmographe remains exactly the same for one of these two categories, their operation is however very different.

The plugins managing graphical symbols constitute in fact classes with objects, each plugin/class inheriting from another plugin/class. The mechanisms of the heritage between plugins are not clarified here, but this part of the software is extensively tested. The crucial point is that thanks to this system of classes with objects, families of plugins (layers, shapes, markers, etc) can be defined in a natural way, each corresponding family with a basic abstract class, a standard interface and knowing of the core that allows an easy communication between core and plugins (see figure 7).
described in the following way: channels contain layers, which on their turn contain symbols; an additional level of hierarchy can be added, by gathering the layers in layer-groups. In the graphical representation, the channels are separated, and the layers are superimposed (see figure 8).

The core manages this tree data structure in a generic manner; it is able to assume standards operations such as: cut, copy, paste, delete, add objects, etc. on the whole of the tree. Each one of these operations is coded, and saved in a file stack, thus allowing to preserve the history of the actions and to ensure the operation of the undo/redo commands.

- Channel 01
  - Layer group 01
    - Background layer (spectrogramm)
  - Layer group 02
    - Layer graphics 01
      - Symbol polygon
      - Symbol polygon
      - Symbol marker
      - etc.
    - Layer graphics 02
    - Layer graphics 03
      - etc.
  - Layer group 03
    - etc.

- Channel 02
  - Layer group 01
    - Background layer (signal)

- Channel 03
  - etc.

Figure 8. A score hierarchy

5 Conclusion
The new Acousmographe will be delivered at the end of 2004 and widely distributed to schools. A team will be established to follow the use of the software, then collect and show some of the best realizations. On the other hand, the main part of the software should be offered as an Open Source code to the computer music community. The GRM hopes that further developments and contributions in the professional domain will be possible, as well as exchanges capacities with some of the best commercial software.

References
Delalande, F., and Vidal, J (1972). “Pierre Schaeffer: Partition animée” (two graphical representations) - 16mm movie.

Proceedings ICMC 2004