DRAWING ELECTROACOUSTIC MUSIC

Jean-Baptiste Thiebaut, Patrick G. T. Healey, Nick Bryan Kinns
Interaction, Media and Communication
Queen Mary, University of London

ABSTRACT

The initial stages of creative design often involve sketching. Electroacoustic composition is no exception to this. Paradoxically, the technologies that enable this form of composition provide little support for the sketching process itself. In this paper we first present evidence for the importance of paper-and-pen sketching in current practice and discuss two strategic representational functions it serves: vagueness and ambiguity. Current music programs offer a variety forms of visual representation but offer only limited support for these functions. We discuss the design alternatives that could be used to provide more support for the creative stages of electroacoustic composition. The program Music Sketcher is presented which aims at bridging the gap between the paper-based sketching activity and the final realization of a piece using standard tools.

1. INTRODUCTION

Previous research on the compositional practices of electroacoustic composers has shown that paper-and-pen sketches often –although not always– form an integral part of the process, despite this population being highly technically literate [9]. In this small survey 75% of the composers reported using pen and paper in the first stages of composition. The most commonly reported initial representation of a piece was a drawing (50%). These responses suggest that, like other design tasks, the initial stages of composition do not involve commitment to specific sound patterns but rather a task where more abstract representations or concepts of a piece provide the starting point. For 13 composers (40%) this stage corresponds to a representation of a rough musical structure or states. For 8 (25%) it corresponds to temporal patterns or dynamics. In most cases these concepts are rendered in a visual representation. In contrast to this, only 5 people in the sample develop the first stages of composition using specific sound parameters. 20 participants (63%) reported beginning with only vague details of the final piece while 6 (19%) would represent a high number of details, or all. Finally, 26 (80%) reported that the piece would evolve from its first representation.

These results for the initial stages of composition suggest processes similar to those described for design in diverse activities such as architecture, playing games, sketchmaps or designing kitchens (see e.g. [19, 8, 10]). Importantly, there is evidence that some of the benefits of sketching derive directly from their relative ambiguity and vagueness, which has been observed in both product design and architectural sketches (see [10] and [8]).

In these empirical studies, there appears to be a relatively straightforward mapping between the sketch space and the domain. For example, spatial extension on the page corresponds in a relatively direct way to spatial extension in the world. Here we focus on a more abstract domain, music, in which the key dimensions - such as aesthetic and temporal structure - have a more obscure relationship to the sketch space. Amitani and Hori argued that externalization in a two-dimensional space efficiently supports music composition creativity by enabling changes in the representation of information ([1]), but did not extend their research to sketches. Music sketches appear to be highly idiosyncratic, as it can be observed in the collection of music sketches published by John Cage in 1969 [4] or by the sketches we received during the survey. In contrast, the graphic user interfaces developed to address compositional issues seem to underuse the potential for ambiguity and vagueness.

For computational applications a clear and and unambiguous semantics is usually considered to be desireable. Graphical user interfaces typically try to minimise ambiguity and in some specialized contexts, graphics have developed into full diagrammatic systems, such as the CSLI Hyperproof system, with rules of well-formedness and clearly defined semantics (see also [12]). Similarly, music software often exploits forms of representation, such as conventional musical notation, which have a relatively well-specified syntax and semantics. These properties are useful where the graphical representations also double as a graphical user interface as, for example, in the music programs UPIC and IanniX, Audiosculpt, MetaSynth or Hyperscore (see section 2 of this paper). However, there is clearly a tension with the apparent creative advantage of the underspecified ambiguous representations provided by sketching.

We begin by considering existing programs that use graphical representations to compose music and how the mapping to sonic events is realized. We then focus on two case studies to justify the design of a new application, which we present in the last part of this paper.

2. EXTERNALIZATION IN EXISTING GRAPHICAL MUSIC PROGRAMS

2.1. The UPIC system

The UPIC[1] system was developed at first as a hardware device that included a drawing table linked to a 64-oscillator synthesizer. The original idea was formed in the early sixties by Iannis Xenakis who composed instrumental music using an architect table to assign pitches and dynamics to the instruments (see e.g. the original sketch for *Metastaseis*[2]). Foreseeing the importance of computers in the development of music, Xenakis pioneered the use of graphical interfaces for music composition by introducing free-hand drawing to control sonic events with the UPIC in 1976. The UPIC system allowed control of the pitch and dynamics of a synthesized waveform. On the main drawing board, time is represented on the horizontal axis, while the vertical axis controls pitch. An intensity envelop can be drawn separately for each line of the board. A waveform can also be defined separately for each line drawn on the main board. The notes are thus drawn on the main board and their envelopes and timbre can be defined individually, enabling people - in Xenakis' own words - to create something alike a full orchestra. Another feature allows users to change the time scale of the drawn score e.g. it could be rendered in 2 seconds or 1 minute.

From a compositional point of view, the UPIC system present many disadvantages. First, the result is constrained to the timbres that one could define with only 64 oscillators. Although all audible timbres could virtually be synthesized without a limit of oscillators, their drawing represents an obstacle, as complex timbres are very difficult to draw by hand. Second, the UPIC system developed apart from the other music programs that are used nowadays to realize a piece (e.g. Protools, Logic or Cubase), thus the structure of a drawing can not be easily linked to such programs. Third, the straightforward mapping between the drawings and the sonic result prevents multiple interpretations by the system and therefore constrains the drawing to the manner in which the mapping is done. These constraints, as a whole, prevent the UPIC system from being a complete composition tool. Rather, it is used as a sound design tool to create original timbres. We now turn to a more recent program called IanniX, whose development was inspired by the UPIC system.

2.2. IanniX

IanniX is a program developed by the La Kitchen company, whose development started in 2001 with Adrien Lefevre. The initial concerns of this program were to address the question of time, represented in most computer programs on the horizontal axis with a fixed, linear progression. Vaggione [18] argued that the representation of time in computer programs should follow the various time scales

[1] UPIC stands for Unité Polyagogique Informatique du CeMaMu

[2] Xenakis produced the electroacoustic piece *Légende d'Er* with the UPIC

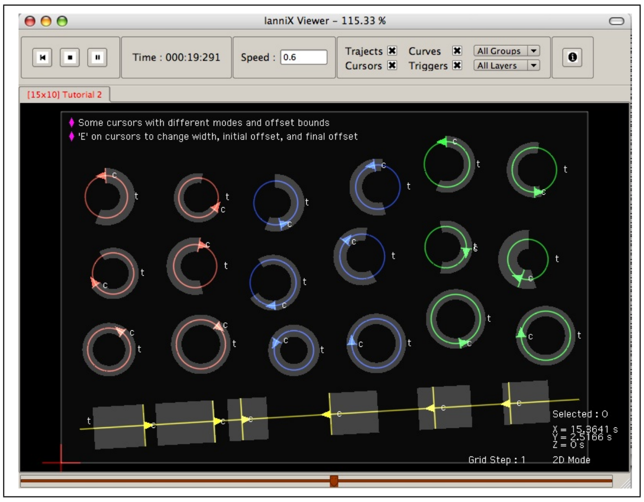

Figure 1. Interface of IanniX, version 0.650b

at which the composer operates ; Dahan [5] argued that horizontal representation of time - which prevail in most sequencers - constrain the construction of parallel dynamic events that could evolve at their own speed rate. To address this problem, IanniX allows people to control multidimensional abstract objects that can be parameterized to run concurrently with various behaviors. An example of these objects is illustrated in the figure 1. Although inspired by the UPIC, IanniX does not offer a manipulable graphic interface and is restricted to the use of circles and lines.

The concurrent time lines enable the construction of parallel sound objects but fails to represent the whole structure, or supporting its elaboration. As we shall see in the case of Metasynth and OpenMusic, a virtual montage room seem to be required to represent the structure. On the other hand, IanniX can communicate with other programs such as Max/MSP through the Open Sound Protocol (OSC). Coupled with other programs, IanniX can support sound design rather than composition.

2.3. Metasynth, AudioSculpt and Sonos

The primary concern of the program Metasynth is sound synthesis. Its interface allows a range of graphical manipulations (including the use of images) to control a frequency additive synthesis. In contrast with the UPIC, Metasynth offers instantaneous audio feedback of the sketch drawn. The graphical functionalities of the program enables people to create original sounds, but these are constrained by the additive synthesis method used to sonify the images. The time is represented on the horizontal axis, while the vertical axis represent the pitch. The vertical axis is scalable, supporting the representation of linear, logarithmic or harmonic scales. But in the end, the user draws more or less a spectrogram which limits the originality of the sounds created, a disadvantage shared with the UPIC system. To balance this, Metasynth offers a montage room which enables people to manipulate recorded sounds in a timeline using the same paradigm as developed in most audio workstations (e.g. Protools or Cubase).

Metasynth offers powerful graphical operations that support sound design by the means of visual creativity. However, it does not support the elaboration of a composition structure by visual means.

AudioSculpt is a program developed by IRCAM. Like Metasynth, it offers the manipulation of a spectrogram, but in a more straightforward way. Several signal processes can be realized on the displayed image such as the filtering of a certain frequencies on a given time window. This program provides an intuitive user interface to alter sounds, but, as with Metasynth does not support composition tasks by the means of visual manipulation. Building on a similar paradigm - the representation and manipulation of a spectrogram - we developed the program Sonos ([15]).

Sonos is a real time application that uses the three color layers RGB to filter and delay a single frame of a spectrogram. In addition, an image could be used to organize the filtering over a fixed period of time. In keeping with the two programs discussed above, the sonic results were limited by the graphical organization of the Fourier transform representation. Although this representation efficiently supports the modification of harmonic contents, it offers little support for a perceptual modification of the timbre or for time based sound processing.

As a conclusion, the spectrogram representation common in the three programs reviewed does not seem to be an appropriate strategy for the support of music composition. This representation, however, contributes to Metasynth's success as a widespread tool for sound design. But composition requires a less constraining environment, such as offered by the program Hyperscore.

2.4. Hyperscore

Hyperscore was designed to simplify the approach to traditional forms of composition. This program facilitates the elaboration of complex structures using a 2-step approach. The first step enables people to define motives and to associate them with given colors. The second step allows people to "paint" a score by the means of the predefined colors, presented in a way similar to a color palette (see Figure 2). The horizontal axis represents time, while the vertical axis allows to controls variations for the motive. The distinction between melodic motives and musical movements is handled in a way that reminds us of the *micro* / *meso* / *macro* distinction, which is an important concept to electroacoustic composition. In Hyperscore, the *micro* scale consists of the smaller elements (MIDI notes), the *meso* scale consists of the motivic scale and the *macro* scale refers to the space where the *meso* motives are combined and organized. The vertical axis represents different constraints in the *meso* space (the pitch of the notes) and in the *macro* space (complex constraints on the harmonic content). This change of representation, also a feature of Diemo Schwarz's CataRT ([11]), facilitates the switching between semantic representations, a feature that

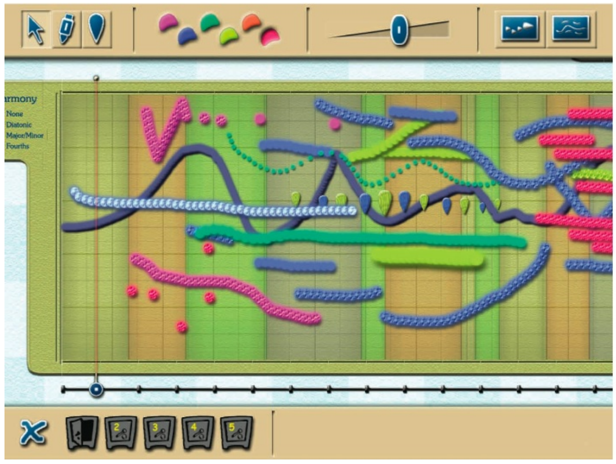

Figure 2. Hyperscore screenshot

is usually commonly found on paper-based sketches (see the case study in [9]).

Hyperscore presents a well defined interface for the drawing of music sketching. In addition the pitch axis which doubles as an harmonic operable space bridges one of the gaps between the paper-based sketching activity and the computer based realization of music pieces. However, as pointed by M. Farbood in [6, 7], this program would improve greatly if it used sound samples rather than MIDI. Moreover, the harmonic constraints of the program limit the results to traditional classical music, which limits musical creativity to specific genres of music. Not surprisingly, most of the compositions made with this program are within the Pop/Rock/Hip hop style, as a result of a highly constrained mapping system.

3. CRITICAL ANALYSIS OF VISUAL STRATEGIES FOR MUSIC COMPOSITION

The programs reviewed above have shown different strategies for the integration of sketching or alternative representations for musical purposes. This review is not exhaustive and other strategies could have been discussed here in more depth such as Open Music ([2]), the Musical Sketch Pads ([13]) or previous work on musical interfaces ([16, 3, 17]). From the representative sample studied, however, we argue that although visual means are at the heart of the design to support music composition, these programs do not address the issues of indeterminacy and vagueness, which are integral to the way paper-based sketches are used for music composition.

Music composition, in particular in the contemporary or electroacoustic genres is nowadays a task that involves a large number of parameters. Composition practices seem to use dedicated programs for different purposes, e.g. the use of several programs akin of Metasynth and/or Max/MSP for sound design and Protools or alike for their organization. The composer Ludger Brümmer reported using more than 20 programs for a single composition [3]. The pro-

[3] The composer reported this in a talk where he presented his work at

grams UPIC, Metasynth and Hyperscore presented above attempt to address all the parameters used in a given musical genre and aim at providing a complete tool for composition. Although it would require further research to assess the variety of music programs used in electroacoustic composition, we propose that an all-in-one approach based on a single representational format cannot give satisfactory results at every level. For example, UPIC and Metasynth would rarely be used to compose a piece but rather be used for the sound design of part of a piece. Hyperscore would not be used outside of the prototyping of traditional classical music pieces or Pop/Rock/Hip Hop songs. Generally, it would not be used by professional composers. In contrast, Audiosculpt and IanniX are focused on specific tasks, which facilitates their integration in the composition chain. IanniX, in particular, does not produced sounds itself but is intended to communicate with other programs, and can even handle the communication with several programs concurrently (such as Max/MSP, Pure Data or Supercollider).

These criticisms lead us to three observations about the design of musical applications. First, visual support for music creativity could be enhanced if initial representations could be successively re-interpreted; for example a representation of pitches over time could be re-interpreted as a spatial locations over time, or as a timbre map. This is consistent with the kind of re-interpretations of sketches of composition seen in [9] and with the kind of figure-ground reversals seen for sketches in other domains (see. e.g. [14, 10]. The underspecified semantics of paper-based sketches suggests an equivalent feature in music programs ; a certain degree of ambiguity or vagueness that supports the exploration of musical structure. Second, the high level constraints inherent in *intelligent mapping* may impose too precise a context, such as e.g. traditional classical music for Hyperscore or harmonic sound synthesis with Metasynth. The mapping between graphic events and sonic events must remain a feature of music programs but the way it operates should be left to the user, in order to avoid predefined effects and support a variety of associations. Third, the role of a program in the music chain must be clearly defined. As we argued above, the numerous strategies to create sounds and organize them would rather be combined than united in a main program that would over simplify the composition process. Besides its role in the music chain, an important feature to consider is how a program is able to communicate information with other programs. A program should be able to import *and* export data, as well as dealing with real time flow of information both at its input and output. We now turn to the presentation of the Music Sketcher.

4. MUSIC SKETCHER: PRESENTATION OF THE PROTOTYPE

Music Sketcher is a prototype for music composition which draws on the three observations we have made in the previous section. In this section we first describe Music Sketcher and then discuss it in terms of our three observations.

Concordia University, Montreal, in November 2006.

4.1. Design of a vague interface

Vagueness, as we pointed out earlier, is an important feature of sketches as it facilitates the development of ideas without premature commitment to the constraints inherent to their realization. While drawing, the composer does not –in general– focus on the specific tools she wants to use but rather on higher level concepts, often of a structural nature.

Of course, not all aspects of paper-based sketches are relevant for this process. The details of texture, the feel of the paper, the fact that it can be crumpled do not seem to be essential. Moreover, drawing on a computer is only loosely analogous to drawing on paper although graphic tablets offer a more intuitive control of drawing than mouse based input. The most basic issue for user-interface design is that the user understands the mapping between the gesture and its representation on the screen. Another useful feature of graphic tablets is that they are sensitive to pressure, which can be exploited for the automated assignment of sound parameters (see the following section).

Music Sketcher thus exploits the analogy to paper sketches but does not try to reproduce the same control over the drawing. We do not expect to obtain the same results with Music Sketcher as we observed in paper drawings, rather Music Sketcher attempts to reproduce some of the key function of sketching. One potential problem with the analogy is that programs have different affordances to paper. By engaging with a computer program, composers may naturally expect a process of action/reaction where the limit to their actions is limited to the set of possible reactions. In comparison, a sheet of paper does not react - although it has physical properties that we can interact with. This difference of approach - *reaction vs reflection* - leads the user to be driven by the possible results rather than by the process itself of *doing*. The implementation of the Music Sketcher is driven by these observations and its interface aims at being as unconstraining as possible.

The prototype presented hereafter consists of a Graphic User Interface developed with Processing and a sound engine developed with the environment Max/MSP. Audio processing was developed using the MSP library of objects.

The drawing interface is represented in figure 3. Drawing on this surface can be done with the mouse or - preferably - with a graphic tablet. The drawing itself has no preconceived mappings, i.e. the vertical and horizontal dimension do not correspond to specific sound parameters. These are to be defined by the user. However, in the version presented in figure 3, the horizontal axis has been set to represent time, whose management is explained in the next section. Layers allow the user to draw a series of sonic event that could be globally manipulated. Manipulations include the rotation of the whole drawing and the translation of single strokes. The user interface provides a framework where the actions are not driven by

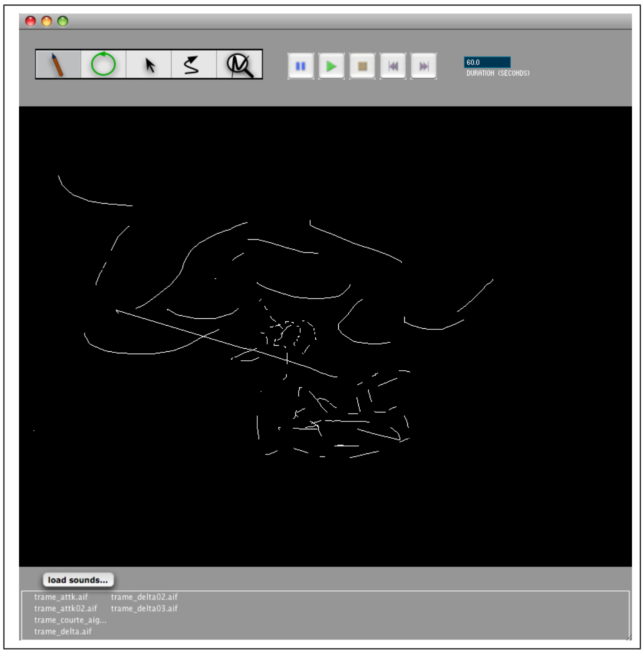

Figure 3. Drawing interface of Music Sketcher.

any expected result, instead, our aim is to enable sketching without constraint. This goal, however, can not be entirely reached on a computer screen at this stage; we discuss what could support this goal in the conclusion section. The next feature at the heart of our design is how the mappings of time are addressed.

4.2. Mapping Time

The mappings between symbols, icons or drawings and sound production constitutes one of the most important debates in the design of music programs. It not only concerns the ease of use of a program but also the possibilities a program can offer. In the case of the program Hyperscore, for example, we argued earlier that the harmonic rules defined for the mappings constrain the results to restricted music genres. The representation of time also constitutes an issue which is addressed in a very similar manner from one program to an other. Probably derived from the Western writing and music scoring, most programs present a time line where sound events are to be interpreted from left to right (see e.g. fig. 2 and 7). It is questionable whether this representation is the most suited for music programs. In this regard, the program Open-music ([2]) offers an advanced development for instrumental music with a various time scale representation called *maquette*. Although the example given below draws on the left-to-right paradigm for didactic reasons, the Music Sketcher aims at offering alternatives for the representation of time which are discussed below.

4.2.1. The drawing-to-sound mapping

At first, and in order not to constrain the drawing, there is no sound related to the points and lines drawn. When

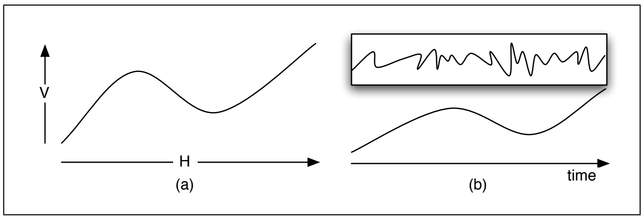

Figure 4. Mapping of a curved line. (a) represents the drawing, and (b) represents the mapping of the evolution of the curve on the vertical axis (V) onto the sound sample.

the user wants to explore possible musical results, sounds can be related to points or lines by dragging a sound file from the list below the drawing surface to a stroke. The velocity at which a stroke has been drawn determines the duration of the sample to be played. The total duration is calculated from the sum of the segments that constitute the entire stroke. It is also possible to calculate the duration depending on the horizontal or vertical progression. Although in this case, the stroke drawn has to be constrained so that there can not be two vertical points corresponding to a single moment in time. A sample affected to a stroke can either be time stretched to fill the equivalent duration of a stroke, or looped (or cropped). If the chosen sound has a longer duration than the length of a stroke, the sound can be played faster ; in the contrary the sound can be played slower. Different mappings than related to time stretching could be envisaged.

The axis which is not used as a time axis provides a series of data that can be used to drive a sound process. The successive coordinates of the drawing are stored in a buffer and interpolated to be used at the time the sample is played (see figure 4). The minimum and maximum values can be adjusted so as to match different audio processes.

4.2.2. The issue of time

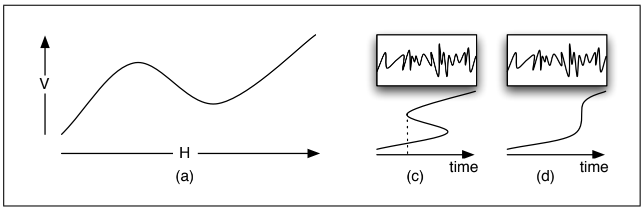

Figure 5. Mapping of the curved line (a) when the vertical axis (V) is chosen to represent time. The dashed line in (c) shows how two points can correspond to a single moment in time. (d) shows how the interpolation is calculated to address the problem.

In the figure 4 we chose the horizontal axis as the time line. If we had chosen the vertical axis as the time line, we would have face the paradox of having several points in the curve to be mapped at the same time (see figure 5). When this situation is observed, we choose the lower point from all the possible points and recreate a new line. Various strategies could be used to address this problem of representation. For example, we could create a discontinuous set of data by choosing randomly (or sequentially) one of the possible points. As mentioned before. we developed an alternative solution to this problem: the time of an event is independent from its length, but linked to the velocity at which it has been drawn. The starting point is calculated depending on the global time axis. For example, if the whole sketch is drawn from left to right, the extremity of a stroke that is the further at the left will constitute the starting point of an event. The left extremity is considered as the trigger of our virtual sequencer. Noticeably, this representation is not homogeneous: two strokes of equal visual importance might be interpreted in very different manners. This is consistent with the way paper-based sketches are used: two similar strokes drawn on paper can represent different sonic events.

In fact, issues of time also occur at the larger scale. This appears to be one of the most constraining aspects of time line based audio programs. A uniform representation of time does not necessarily match the composer's perception of successive musical moments. For example, a complex musical articulation that elapses in a short period of time likely requires a great attention to detail. This can be addressed by a non linear representation of time. In the Music Sketcher, this non linear representation can be controlled through the scaling of a line. This line controls the scale of the representation and the speed rate at which the drawing is "read". The representation of non linear controller resembles to the representation of audio envelopes: when the line is close to the top the time span is faster. Inversely, when the line goes toward the bottom, the time span is slower .

4.3. Example of mapping: Granular synthesis

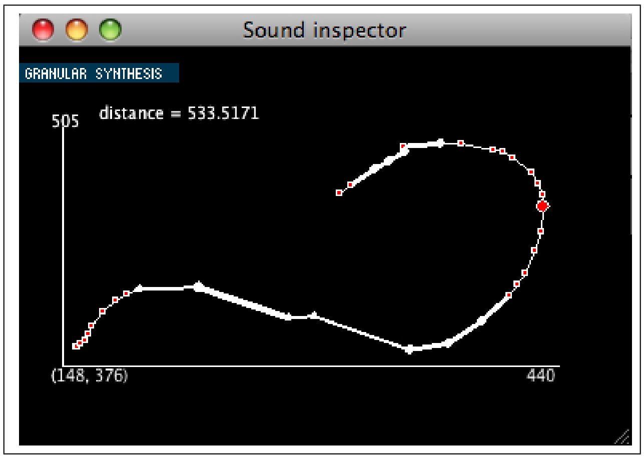

Figure 6. The sound inspector: this interface allows to control precisely the evolution of parameters.

A first prototype of the Music Sketcher allowed to control the pitch of sound samples using the vertical values of a stroke. This process is rather intuitive, drawing on the tradition of score-based notation. However, the sonic result was simplistic and inaccurate, in comparison with existing audio processes dedicated to pitch shifting. In comparison, granular synthesis offers many advantages for the testing of our system. First, this synthesis method builds on existing sounds and generates complex results. Second, in contrast with most synthesis methods, it requires few, high level parameters such as the density, the duration and the location of the grains. The density is controlled by the speed of the gesture: the faster a segment has been drawn the higher the density. The duration of a grain is controlled by the successive values on the vertical axis. The location of the grain to be synthesized from the sample is determined by the horizontal axis: the extreme right point corresponding to the end of the sample. Finally, the width of the stroke controls the amplitude of the grains.

The *sound inspector* represented in figure 6 shows a simple stroke whose segments and nodes can be changed, enabling to control more precisely each parameter. Each node can be dragged to a new position. The width of a segment can be thickened or thinned using the mouse. Although granular synthesis seems appropriate for the study of our interface, more mappings are envisaged in the future. However, the focus of our research remains on the user interface and how the data created could be shared with other programs rather than a study of the mappings.

4.4. The role of Music Sketcher in the music production chain

Instead of attempting to solve all composition issues in one program, Music Sketcher focuses on the first stages of composition. As such, the goal of this program is to develop musical ideas that are imagined as drawings rather than to compose a piece from beginning to end. It aims at communicating with other programs, following the example of Iannix or OSC ([20]). Audio workstations present several advantages for the realization of musical pieces that this program will not attempt to compete with. The export of the sonic events drawn to MIDI events or audio files in a digital audio workstation is currently studied.

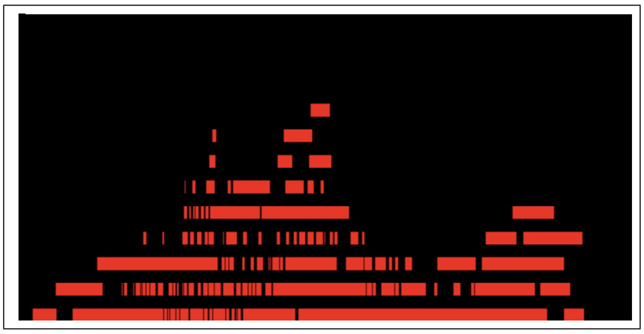

Figure 7. Standard time line representation of the drawing of fig. 3

The audio events drawn in figure 3 correspond to the timeline represented in figure 7. It illustrates how the sound events are likely to be manipulated in standard workstation once they have been designed with the Music Sketcher.

5. CONCLUSION

Building on an empirical study and on a review of existing music programs, we have observed that the early stages of electroacoustic music composition involve activities which are not addressed by standard music programs. These studies led to three observations. First, vagueness is integral to the ways composers get started with music composition ; this vagueness often being expressed in the form of a sketch. Second, the *intelligent mappings* developed in some advanced applications constrain the results to specific contexts, which, regardless of their pedagogic roles, do not support electroacoustic composition. Moreover, as observed in most music programs, the linear representation of time does not match the perception of musical time. Third, the role of a program in a music composition process must be defined in order to match the various stages composers go through while they are composing. We have presented the prototype of a program which intends at answering these observations. We focused in particular on the development of a drawing interface where sounds are not typed, and whose representation of time varies for each sound and for the whole structure as well. Future work will focus on evaluating the strategies developed for the Music Sketcher. The analogy of music sketches developed in this tool will no doubt affect the process of composing itself - a future study will examine to what extent the Music Sketcher addresses the observations that motivated its development.

6. REFERENCES

[1] Amitani S. and Hori K. “Supporting musical composition by externalizing the composer’s mental space”, *Transactions of Information Processing Society of Japan*, 2001.

[2] Assayag G., Baboni J. and Haddad K., “Openmusic 4.0 user’s manual reference”, *IRCAM, Paris*, 2001. www.ircam.fr/openmusic.

[3] Bokesoy, S., Thiebaut, J.-B., “An approach to visualization of complex event data for generating sonic structures”, *Proc. of ICMC*, 2006.

[4] Cage J., “Notations”, New York: Something Else, 1969.

[5] Dahan K., “Domaines Formels et Representations dans la Composition et l’Analyse des Musiques Electroacoustiques”, *Ph.D. thesis*, CICM, Université Paris VIII, France, 2005.

[6] Farbood M., Pasztor E. and Jennings K., “Hyperscore: A Graphical Sketchpad for Novice Composers”, *Computer Graphics and Applications, IEEE*, pp. 50-54, 2004.

[7] Farbood M., Kaufman H., Jennings K.,“Composing with Hyperscore: an intuitive interface of visualizing musical structure”, *Proc. of ICMC*, 2007.

[8] Goldschmidt G., “The dialectics of sketching”, *Creativity Research Journal*, 4(2):123–143, 1991.

[9] Healey, P. and Thiebaut, J.-B. “Sketching Musical Composition”*Proceedings of the Cognitive Science Society conference*, Nashville, USA, 2007.

[10] I. Neilson and J. Lee. “Conversations with graphics: implications for the design of natural language/graphics interfaces”, *International Journal of Human-Computer Studies*, 40:509– 541, 1994.

[11] Schwarz D., Beller G., Verbrugghe B. and Britton S.,“Real-Time Corpus Based Concatenative Synthesis with CataRT” *Intl. Conf on DAFx*,2007

[12] Stenning, K. and Oberlander, J. “A cognitive theory of graphical and linguistic reasoning: Logic and implementation”, *Cognitive Science*, 19:97-140, 1995.

[13] Subotnik M., “Musical Sketch Pads online activity”, http://www.creatingmusic.com/mmm, 1999-2008.

[14] Suwa and Tversky B., “What architects see in their sketches: Implications for design tools”, in Canada Vancouver, British Columbia, editor, *Conference on Human Factors in Computing Systems*, pages 191–192, 1996.

[15] Thiebaut J.-B., ”Visualisation du son et réversibilité, l’exemple du logiciel Sonos”, *Proc. of JIM 05*, 2005. Sonos is available at http://www.dcs.qmul.ac.uk/ jbt/sonos/.

[16] Thiebaut J.-B., ”Pompiloop, logiciel de création musicale jeune public”, *Proc. of JIM 05*, 2005.

[17] Thiebaut J.-B., Bello J. P. and Schwarz D., “How musical are images? From sound representation to image sonification: an eco systemic approach”, *Proc. of ICMC*, 2007.

[18] Vaggione H., “Some ontological remarks about music composition processes”, *Computer Music Journal*, pp 54-61, 2001.

[19] van Sommers, P., “Drawing and Cognition - Descriptive and Experimental Studies of Graphic Production Processes”, Cambridge University Press, 1984.

[20] Wright M, Freed A., “Opensound Control: a new protocol for communicating with sound synthesizers”, *Proc. of ICMC*, 1997.