

SYMBOLIC CONTROL OF SOUND SYNTHESIS IN COMPUTER ASSISTED COMPOSITION

Jean Bresson

Ircam

Musical Representations Team
Paris, France

Marco Stroppa

Staatliche Hochschule für

Musik und Darstellende Kunst
Stuttgart, Deutschland

Carlos Agon

Ircam

Musical Representations Team
Paris, France

ABSTRACT

This paper presents current works and future directions concerning the control of sound synthesis in OpenMusic. We will particularly focus on the concept of synthesis models for composition, and on the representation of sound synthesis objects.

1. INTRODUCTION

Computer-assisted composition (CAC) systems are designed to allow the use of computers for formalizing and experimenting with musical ideas, for creating and manipulating musical structures through programming techniques. The progress of computer-music research in the field of sound synthesis extended the possibilities for sound generation, allowing the composers, following the pioneering work of K. Stockhausen and I. Xenakis, to dig further than instrumental music, into the proper composition of sound. However, despite the many software synthesizers at hand, they usually remain hard to control without significant technical skills. In this article, we will try to propose solutions to integrate sound synthesis and compositional systems, in order to allow for real musical expressiveness by means of synthesis technologies.

OpenMusic (OM), is a visual programming language dedicated to music composition [1]. Following some preceding articles concerning the implementation of high-level data structures for controlling sound synthesis in OpenMusic [2][3], we will present here some tools for the representation of synthesis data and processes, and means to link compositional concepts with sound synthesis in this environment.

After a general presentation of this issue, we will put forward our approach of the representation of sound models with some examples in OM.

2. SOUND SYNTHESIS AND COMPUTER ASSISTED COMPOSITION

Although CAC environments are generally used for writing instrumental music, and manipulating symbolic structures (such as notes, chords, rhythms, etc.), their functional paradigm can be conceptually applied to the realm of the composition of sound. However, the control of sound synthesis also brings to CAC some specific problems.

2.1. New problems of representations

The first obvious specificity of sound synthesis in CAC environments is the great amount of data that needs to be computed. However, another really specific aspect stands in the nature of this data, generally made of control functions or other kind of time-sampled data, for which the basic elements may not have any significance outside of their context, contrary to the symbolic objects of CAC systems. We call them sub-symbolic data.

In writing an instrumental score, the physical reality is not entirely specified, but just expressed until a certain level. When synthesizing a sound, however, the sound must be thoroughly described. Due to the potentially important amount of parameters that ought to be as little constrained as possible, the issue of its description and notation is therefore more complex. In order to achieve a more efficient computation, and an easier user interaction, we may be tempted to curb the amount of parameters that can be freely controlled, and fix the others, but it is important to let the user make this choice and to provide him with means to do it. Moreover, the problem of having pertinent subjective representation of these parameters still remains unsolved. The system cannot decide automatically how to interpret data in order to keep a correspondence between the sonic and musical fields, but it should allow the user to do so.

In spite of some potentially powerful systems, the earliest attempts of musical control of sound synthesis were not widely used by composers, probably because of their low level, generally text-based, interfaces. A meaningful, musical, and first of all subjective representation of synthesis data seems to be mandatory. Some studies investigated the relationship between visual and perceptual descriptions of synthesis parameters (e.g. [13] [7]) but they are either hard to be put in practice in a given synthesis environment, or require important restrictions.

2.2. A specific conception of time

One of the most innovative characteristics of sound synthesis, at both a musical and technological level, is also that it makes it possible to connect the microscopic and macroscopic compositional aspects of music (i.e. the creation of the inner structure of sounds and the composition using these sounds). One might then assume the necessity of a variable scale of temporal and logical granularities.

3. SYNTHESIS MODELS: A MEANINGFUL REPRESENTATION OF SOUND OBJECTS AND PROCESSES

3.1. Models for composition

When using computers, an important part of the composer's task is to formalize her/his musical ideas. An interesting approach of this compositional formalization is the concept of models as described by M. Malt in [10]: a compositional model is a conceptual representation that is a link between an abstract musical concept and the concrete world. A CAC environment must then be a place where composers develop and manipulate models, by means of experiments on data and processes. These models should be connected together, interact, be embedded in other models, as are the corresponding musical concepts in the composer's mind.

3.2. Synthesis models

In sound synthesis we are used to distinguish various families of models, corresponding to different synthesis methods (physical models, abstract models, signal models, etc.) However, one tends to identify the models to these techniques, but this not enough to define real compositional models. Such a definition must also reflect a musical intention, that is, it must implicitly determine an identifiable group among the infinity of possible sounds. What really defines a model might be the choice of the variable and invariant parameters, of the degrees of freedom, of the deterministic or non-deterministic parts of the algorithm. From this perspective, the choice of the synthesizer can be just a possible variable parameter in a data-based sound model.

A sound model therefore represents a sonic potential, a multidimensional space that a composer can experiment and manipulate in order to explore a class of sounds determined by this model. The resulting sound is a realization, an instance of this sound potential. It will be more or less musical and lively depending on the richness of the model.

3.3. Sound representation

From the perspective put forward above, a compositional object corresponding to a physical sound can no longer be a simple sound file or waveform data buffer, but a set of data and processes yielding a sonic result, consistent with all the possible objects contained in the model. The external representation of this model gives it a musical potential: it determines the realm of modifications and possibilities of experimentation [6].

A complete representation requires the use of a symbolic description language, which should incorporate different levels, that is: the representation of the processes of sound creation that define the model; the representation of the processing parameters and data that are the compositional inputs of the model; and the representation of the real resulting sound.

4. SYNTHESIS MODELS IN OPENMUSIC

The OpenMusic Lisp-based environment can communicate with external synthesizers by creating parameters files or structures and sending commands to external programs. Composers can therefore use the symbolic and computational possibilities of OM in order to control these sound synthesizers. In this section we present examples of such applications.

4.1. Sound transformation with SuperVP

The OM-AS library, written for OpenMusic by H. Tutschku, contains a set of functions that create parameter files for SuperVP [5], a phase vocoder whose re-synthesis of sound files can be modified by intermediate, possibly time-varying, transformations such as time stretching, pitch shifting, filtering, etc. A parameter file allows for complex time-varying functions to be specified. The new OM-SuperVP interface computes the sound files within an OM patch (see Figure 1), so that sounds can be created directly using graphical algorithms.

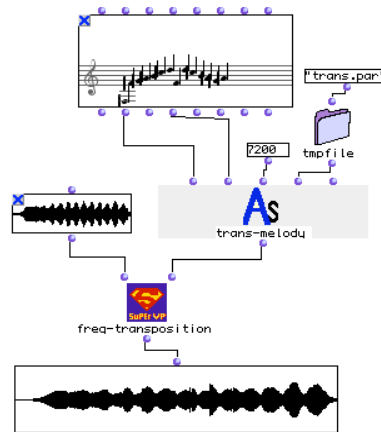


Figure 1. Sound synthesis by SuperVP in OpenMusic. A sound is transformed by a time-varying frequency shift computed from a melody.

4.2. Csound synthesis patches

The library OM2CSound was originally written in PatchWork [9] by L. Pottier. It allows the graphical design of Csound [4] score files. The library was then ported to OpenMusic and enhanced by K. Haddad who developed boxes for generating Csound orchestras. A Csound synthesis function can now synthesize the sound in OM. Figure 2 shows a simple example of a Csound synthesis patch.

This approach is a functional generative model for sound description, which potentially allows to produce a large set of sounds (Csound can be used to implement many kinds of different synthesis techniques). It also illustrates the potential of a graphical implementation of synthesis processes within a visual language. The inner structure of the process becomes more accessible and can be easily edited.

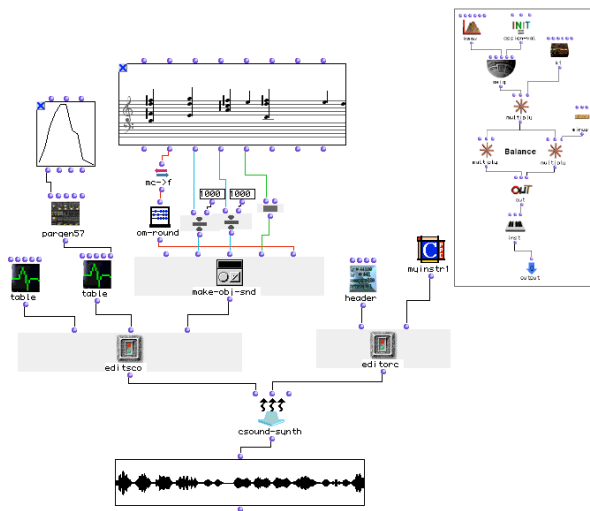


Figure 2. A simple Csound patch in OpenMusic. Double-clicking on the rightmost *myinstr1* box opens the graphical definition of the instrument (at the right).

However, a graphical patch cannot contain as much information as a textual score or orchestra file does. The production of complex sounds therefore requires describing abstractions or partially defined models in order to limit the amount data and allow for musical experimentation.

4.3. OMChroma : high level abstractions

The Chroma system [12] has been developed by M. Stroppa during his musical activity. The concept of "virtual synthesizer" allows the composer to control sound synthesis by creating data sets which are internally translated and formatted for different possible synthesizers (Csound, CHANT, etc.) OMChroma [2] is the implementation of Chroma in OpenMusic (see Figure 3).

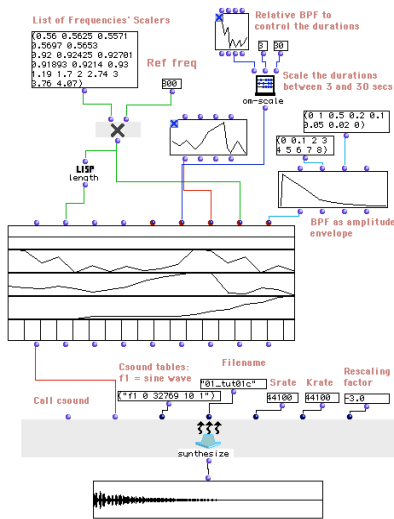


Figure 3. Csound additive synthesis in OMChroma. The matrix can be instantiated in different ways (functions, bpf, numerical values, lists), and then is formatted for the synthesizer by the *synthesize* generic function.

This system provides pre-determined matrix classes that represent different synthesis models and can be subclassed and enriched by the user. The object-oriented system of OpenMusic is used to improve the modularity and the behaviour of these classes during the computation of the matrix's components and the synthesis process. These high-level abstractions reflect the conception of data-processing objects corresponding to classes of musical objects sharing common properties.

This matrix-based representation of a sound thus permits powerful computational possibilities, but its unfolding in time still remains a problem. In [3], we proposed a generalized sound-description model based on SDIF (sound description interchange format), but the issue of a meaningful representation and manipulation of synthesis parameters could only be partially addressed.

4.4. Extension to other synthesis models

Being a complete programming language, OpenMusic allows to implement any kind of synthesis model in it.

The Modalys physical-model synthesis has recently been implemented in OM by N. Ellis at Ircam.

The communication with external systems is also possible via OSC. Control parameters can then be generated for any synthesiser compatible with this standard transfer protocol.

5. TEMPORAL ISSUES

5.1. Synthesis models in time

Time in composition must be considered at multifarious levels : as linear time, final arrangement of pre-calculated musical material, but also as logical and hierarchical time, with temporal dependences, recursively-nested structures, rules and constraints between the different objects.

The examples presented in section 4 do not have any advanced temporal structures. We could see them as a way to generate sound objects "out of time", or rather with only a local time. These objects can be considered as compositional primitives [8] that need to be integrated in a musical structure. The next step consists in embedding these primitives in a temporal context in order to unfold the models in time.

5.2. Synthesis models in the *maquette*

In OpenMusic, the *maquette* [1] is a special object which is at the same time a scheduling tool that can play MIDI and audio objects, and a patch (i.e. a graphical program) in which these objects can be manipulated, connected and computed.

A *maquette* contains boxes which can represent different types of objects in a global temporal context. Such objects are either simple musical objects (sounds, notes, chords, etc.), patches (graphical programs having a resulting "temporal" output), or other *maquettes*. In this way, hierarchical structures can be created, as well as

complex functional and temporal relationships between the patch boxes. Synthesis models patches as those examined in section 4 can be embedded in this temporal structure, which could integrate their local temporal logic within the general time flow.

We made an example with the Marco Stroppa's piece *Traiettorie... deviata* [11]. The electronic part of this piece was originally computed at the Centro di Sonologia Computazionale of the University of Padua (Italy) in 1982, using an ancestor of the Chroma system. A part of the electronics was recently re-created by the composer in OpenMusic. The sound components were created with patches similar to the patch in Figure 3. However, the original temporal organization was a hierarchical structure of such sound units that could only be reproduced in OMChroma by the algorithmic concatenation of matrices, shifted with the corresponding hierarchical offset. By putting these patches in *maquettes* (see Figures 4 and 5), we could recreate the temporal and hierarchical structure of the piece.

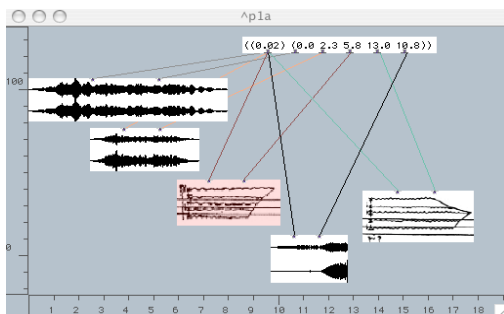


Figure 4. Sound synthesis patches in a maquette. Each box contains a patch producing a sound.

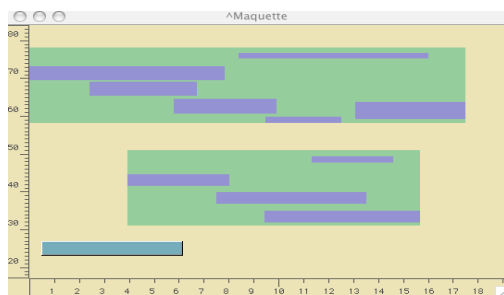


Figure 5. Top level structure of an extract from *Traiettorie... deviata* (M. Stroppa) reconstructed in a maquette. The topmost box is the maquette of Figure 4.

The subcomponents of the *maquette* of Figure 4 are displayed either as sounds, or as hand made pictures similar to the original "paper" score. The sound boxes are calculated with the input data from the topmost box. Opening each of these boxes allows seeing their contents either as sounds or as synthesis patches.

6. CONCLUSION

In this discussion about the symbolical control of sound synthesis, we emphasized some of the salient features required for such an environment: an expressive, personalized, interactive representation of sound synthesis objects, and means to implement synthesis

models that take into account the temporal organization and user-defined rules at different structural levels.

We believe that the association of visual programs with the properties of the *maquette* might allow to integrate the different successive layers of electronic composition: from the sound creation to the global structure construction. The sonic representation and temporal organization allow an easier interpretation of the composer's thought, as well as the possibility to experiment upon the musical piece itself.

This graphical and interactive system may also be a way for documenting, transmitting, analysing and learning about a compositional thought, by means of formalization and representation of the models. It could therefore lead to the development of a common knowledge on sound synthesis for musical composition.

7. REFERENCES

- [1] C. Agon "OpenMusic : Un Langage Visuel pour la Composition Assistée par Ordinateur", PhD. Thesis, Université Paris VI, 1998.
- [2] C. Agon, M. Stroppa, G. Assayag "High Level Musical Control of Sound Synthesis in OpenMusic", *Proc. ICMC*, Berlin, 2000.
- [3] J. Bresson, C. Agon "SDIF Sound Description Data Representation and Manipulation in Computer Assisted Composition", *Proc. ICMC*, Miami, 2004.
- [4] R. Boulanger (ed.), *The Csound Book*, MIT Press, 2000.
- [5] Ph. Depalle, G. Poirot "A Modular System for Analysis, Processing and Synthesis of Sound Signals", *Proc. ICMC*, Montreal, Canada, 1991.
- [6] G. Eckel, R. Gonzalez-Arroyo "Musically Salient Control Abstractions for Sound Synthesis", *Proc. ICMC*, Aarhus, Denmark, 1994.
- [7] K. Giannakis, M. Smith "Auditory-Visual Associations for Music Compositional Processes: a Survey", *Proc. ICMC*, Berlin, 2000.
- [8] H. Honing, "Issues in the Representation of Time and Structure in Music", *Contemporary Music Review*, 9, 1993.
- [9] M. Laurson, J. Duthen "Patchwork, a Graphic Language in PreForm", *Proc. ICMC*, Ohio State University, USA, 1989.
- [10] M. Malt "Concepts et modèles, de l'imaginaire à l'écriture dans la composition assistée par ordinateur", *Actes du séminaire Musique, instruments, machines*, 2003.
- [11] M. Stroppa. *Traiettorie* (1982-84), a cycle of three pieces (*Traiettorie... deviata*, *Dialoghi*, *Contrasti*) for piano and computer-generated sounds. Recorded by Wergo: WER 2030-2, 1992.
- [12] M. Stroppa "Paradigms for the High-level Musical Control of Digital Signal Processing", *Proc. DAFX*, Verona, 2000.
- [13] D. L. Wessel "Timbre Space as a Musical Control Structure", *Computer Music Journal* Vol. 3(2), 1979.