TimeSculpt in OpenMusic

Karim Haddad

Abstract. *In the last decade my work has been essentially focused on musical time. I have used computer-aided composition and most particularly, OpenMusic. This article deals with compositional strategies and musical duration, and skips considerations regarding pitch in order to better focus on our subject except when pitch is related to our main disscussion.*[1]

1 How "time passes in OpenMusic ..."

In OpenMusic time is represented (expressed) in many ways. Time could be:

- a number (milliseconds for instance);
- a rhythm tree (an OpenMusic representation of musical rhythm notation [1]);
- a conventional graphical symbol (a quarter note) in OpenMusic *Voice* editors;
- an "unconventional" graphical object (an OpenMusic temporal object).

Another important issue is how musical time is conceived internally (i.e implemented as a structural entity) [2].

Most computer environments and numerical formats (MIDI for instance), represent the musical flow of time broken down into two expressions:

- the date of the event (often called onset);
- the duration of the event (called duration).

We notice already that this conception diverges from our own traditional musical notation system which represents a compact, readable time event + duration. The only reason for this is that generally the computer representation of time is made not with symbols but with digits.[2] That, I believe, is why today's composers should make themselves familiar with a "listing" representation of musical events.

[1] One can argue that these fields are indissociable and most particularly rhythm and duration. We will consider in this article that duration is from a different order than rhythm (think about Pierre Boulez *temps strie* and *temps lisse* in *Penser la musique aujourd'hui* [3] which is a well accepted view nowadays).

[2] Time is sequentially expressed and not iconically symbolized (as a whole entity like a measure containing rhythm figures, having a tempo assignement and a rhythm signature). Through this conception tempo is meaningless.

Since the MIDI standard is integrated in OpenMusic, this representation type is common to most OpenMusic musical objects, and since there are OpenMusic libraries that generate Csound instruments that use this "time syntax", one needs to learn this particular representation in order to deal with these objects accurately, especially when applying them to controlled synthesis.

1.1 `dx->x` and `x->dx`

`dx->x` and `x->dx` are easy to use and are very practical when it comes to duration and rhythm. Starting from a list of duration values in milliseconds and a starting point, `dx->x` will output a list of intervals (duration values) according to these parameters. Vice-versa, `x->dx` will output a sequence of time-dates starting from a list of duration values.

To illustrate this mechanism, we start with a marker file created with Snd. This file represents a list of time events automatically or manually generated. In figure 1, the soundfile was marked manually.

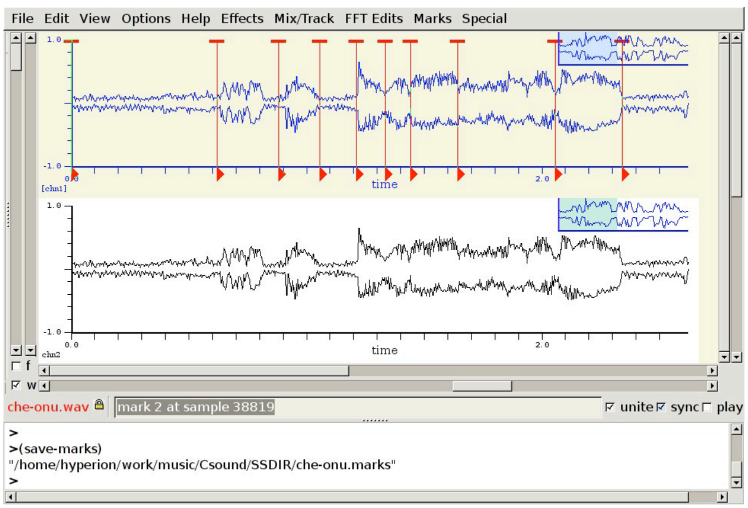

Figure 1. Markers in Snd.

Once the marker file is imported into OpenMusic, we quantify it using `omquantify`, as shown in figure 2.

Careful examination of the rhythmical result may reveal a wide range of duration values (in the present case they could be considered as a series of durations). We might also use it as a sequence of accelerando/decelerando profiles (modal time durations). In either case, it is a rich potential rhythmical material. It can be used to notate symbolically and/or integrate a sound file into a score.[3]

[3]It is also very practical for coordination between musicians and tape.

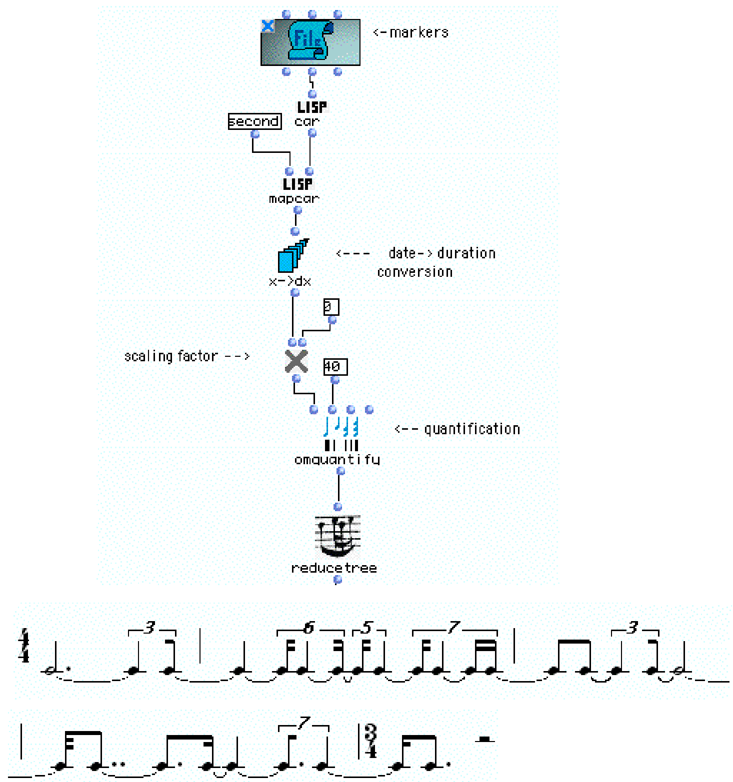

Figure 2. Quantification of markers.

We can of course extend this "symbolic" information and consider it as a compositional material for example, applying to it contrapuntal transformations such as recursion, diminition or extension, simply by multiplication, recursion, permutation.

1.2 Combinatorial processes of rhythmical structures

Another aspect of rhythm manipulation that I use is exactly opposite to the preceding example. Instead of extracting rhythm from a physical source (i.e. such as a sound file) I apply directly combinatorial processes of rhythmical structures using the internal definition of rhythm in OpenMusic called *Rhythm Trees* (RT) [1]. This is a wonderful technique for creating any imaginable rhythm, simple or complex, and since the RT standard is both syntactically and semantically coherent with musical structure, it makes rhythm manipulation and transformation efficient.

It is for this reason I came to write the *Omtree* library, which was basically a personal collection of functions, for OpenMusic. The functions allow 1) basic rhythm manipulations 2) practical modifications and 3) some special transformations such as proportional rotations, filtering, substitution.

The whole stucture of "*...und wozu Dichter in dürftiger Zeit, ...*" for twelve instruments and electronics is written starting from the generic measure in figure 3,

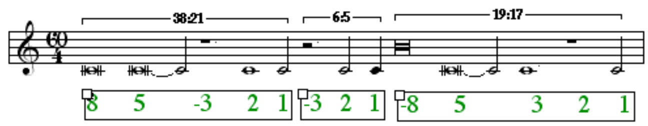

Figure 3. Generic measure of "*...und wozu Dichter in dürftiger Zeit, ...*".

which corresponds to the following rhythm tree:

```
(?  (((60 4) ((21 (8 5 -3 2 1)) (5 ( -3 2 1)) (34 ( -8 5 3 -2 1))))))
```

Rotations are calculated on the main proportions based on the Fibonacci series (rotations of D elements—durations, and rotations on the S elements also—subdivisions) by the patch in figure 5. Rotation number 3 is shown in figure 4.

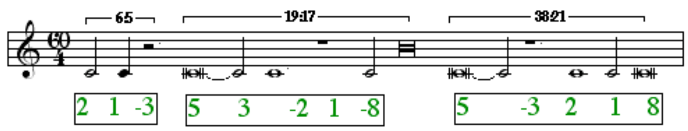

Figure 4. First rotation.

The corresponding rhythm tree is as follows:

```
(?  (((60 4) ((5 ( 2 1 -3)) (34 ( 5 3 -2 1 -8)) (21 (5 -3 2 1 8 ))))))
```

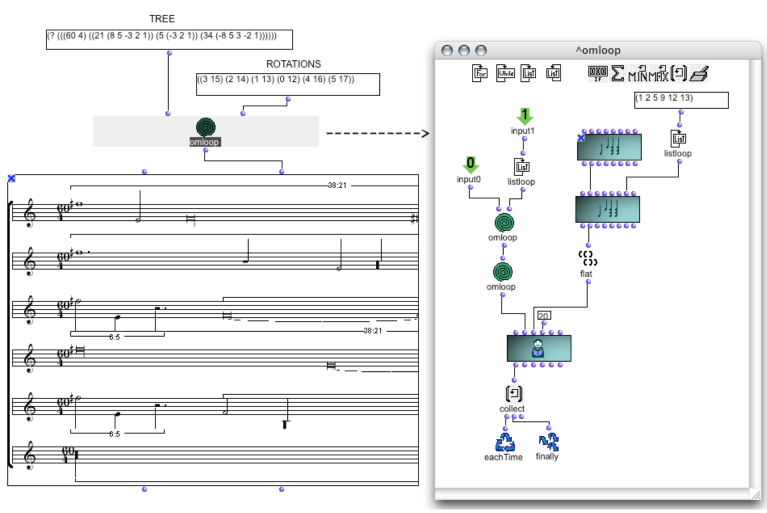

Figure 5. Rotation patch.

The result is a six-voice polyphony. The pitches are also organized in ordered rotation and heterophonically distributed among the six voices (figure 6). The excerpt in figure 7 shows the same result after quantification.

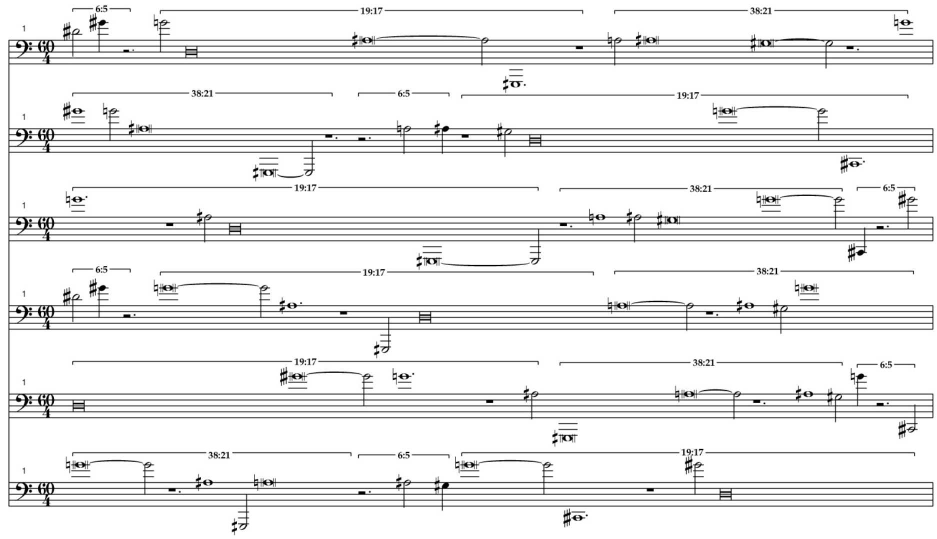

Figure 6. Six-voice polyphony.

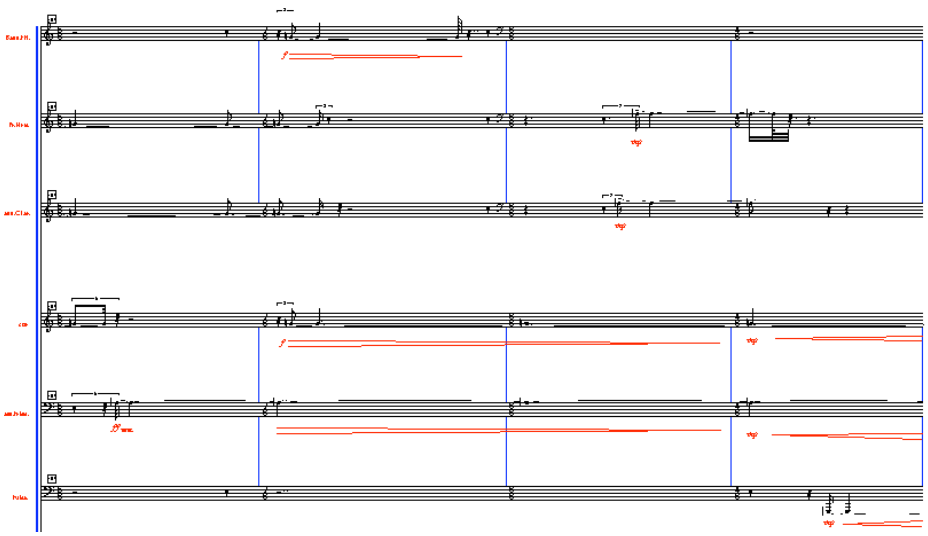

Figure 7. Excerpt from "*...und wozu Dichter in dürftiger Zeit,...*" for twelve instruments and electronics.

2 Topology of sound in the symbolic domain

2.1 Form

We can consider sound analysis as an open field for investigation in the time domain, where interesting and unprecedented time form and gestures can be found and used as basic (raw) material for a musical composition. This new approach is made possible thanks to fast computers. But sound analysis being a vast field of research (in which one can find multiple kinds of analysis and visual or raw data formats), one must take care not to forget the nature of this analysis and its original purpose. Sound analysis could therefore be a rich source from which symbolic data can be retrieved and freely remodelled according to the composer's needs.

Sound analysis data can also be considered as potential correlated vectors. These, depending on the analysis type, can be streams of independent linear data, or more interestingly still, data arrays. The types of data found most often are directly related to the nature of sound. On an abstract level, this can be regarded as a pseudo random flow or considered as coherent interrelated orders of data, again depending on the analysis type chosen.

When using sound analysis as a basis for music material production and most particularly in the time domain, it is important to note that the following approach is not a "spectral" one in the traditional sense,[4] but on the contrary should be considered as a spectral-time approach. Frequency domain will be translated into time domain and vice-versa following the compositional context, as will be shown later in the present article.[5]

This "translation" is possible with the wide variety of analysis types (additive, modRes resonance modes, etc.), not forgetting the many data formats available, whether in visual or numerical form.

The way the material is used depends on the musical project. Different orders of "translations" in the symbolic field can be applied. Form can be literally extracted from the analysis data or taken from symbolic material. The mixed sources (symbolic and analytical) are then fused together in the compositional process, and that is where the tools are very important. OpenMusic is where the analytical and the symbolic come together in a kind of fusion in the field of musical time.

2.2 *No one to speak their names (now that they are gone)*

The structure of *No one to speak their names (now that they are gone)* for two bass clarinets, string trio and electronics, is based on an aiff stereo sound file of 2.3 seconds duration.

Considering the complex nature of this sound file (friction mode on tam-tam), it has been segmented into 7 parts (figure 8). The segmentation is based on the dynamic profile of the sound file.

[4] Meaning that form and strategies are primarily based on pitch.

[5] This was the initial approach in Stockhausen's well-known essay "...wie die Zeit vergeht..." [7].

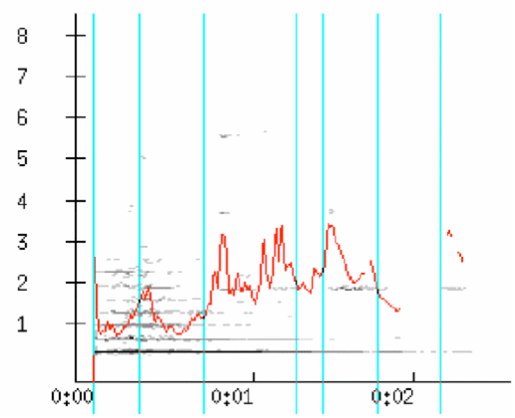

Figure 8. Segmentations.

We may consider a sound as an array of n dimensions (as shown in figure 9) with potential information that can be translated into time information. It seems natural to construct this array using the additive analysis model (time, frequency, amplitude and phase). This is rather a straightforward description that could be used to process the sound directly into the time domain or for eventual resynthesis. Other sound analysis/description is available such as the spectral analysis, lpc analysis, the modRes analysis and so on. Here, the modRes analysis was chosen.

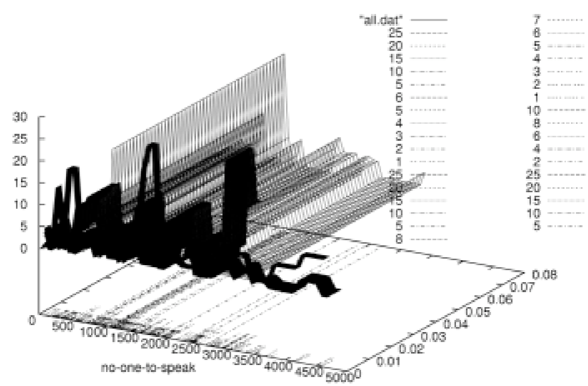

Figure 9. Array of n dimensions.

All the examples described above are discrete windowing analyses, from which the time domain is absent. Most of them have time addressing, but the last one (modRes analysis) is an array of dimension 3 (Frequency, amplitude and bandwidth/Pi) computed by the patch in figure 10.

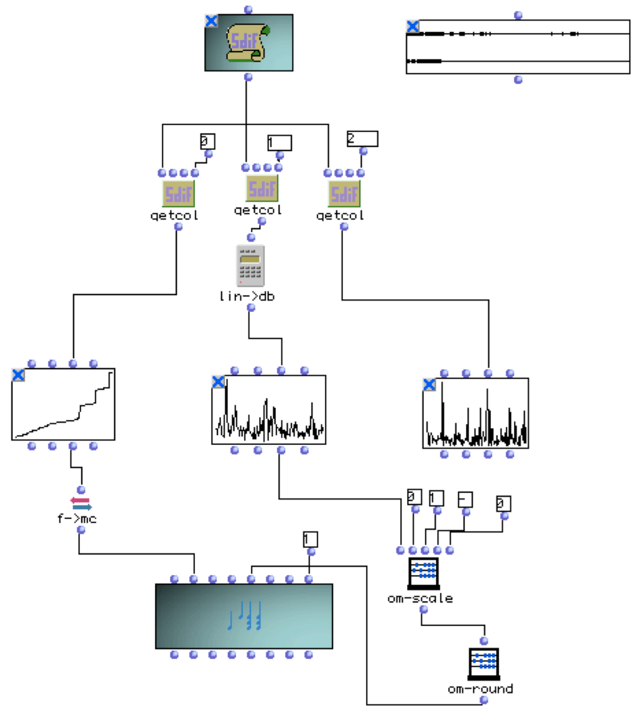

Figure 10. Frequency, amplitude and bandwidth data.

A sound analysis/description and an array of n dimensions can be translated into the time domain from array to array, i.e. the analysis data could be read in any plane, vertically, diagonally, or any combination of arrays. This "translation" is of course arbitrary and is meant to be a translation in the symbolic domain, the score being another kind of array. Although the operation may seem arbitrary, (which indeed it is), in my opinion there are two pertinent points to be considered.

Firstly, (as we will see later) the sound array is processed in a completely interdependent way, taking into account all the proportionate weight relations contained within it. The coherence of the sound resonance will be so to speak "reflected" in the symbolic domain through specific analogical modes (dynamics, durations and pitch), which are not supposed to be literally associated one by one (i.e. exact correspondence of parametrical fields is not necessary). In this piece they are permutated.

The second important point is that this translational strategy establishes a strong relationship between the electronic and the acoustical components of the piece creating strong formal fusion.

Moreover, if we visualize the given data in a three-dimensional graph (see figure 9) we will see many folds ("plis" [5]) of different densities. These are directly related to the polyphonic gesture representing the point-counterpoint technique used in the score.

As we can see in the figure 11, there are two parallel musical processes: the electronic part (tape), which is also entirely constructed with the initial material (the tam-tam

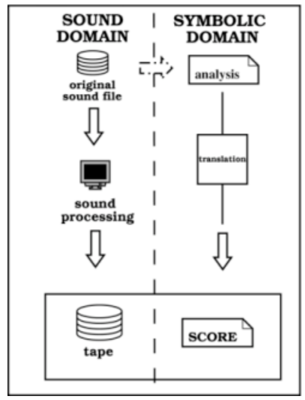

Figure 11. Two parallel musical processes.

sound file), and the score part. The semantic bridge is shown as a dashed arrow. It is through analysis that both domains communicate with each other.[6] In the case of resynthesis, another bridge could be established in the other direction (from symbolic to sound domain) but this is not the case in our present composition.

In *No one to speak their names (now that they are gone)*, using the modRes analysis, the bandwidth array has been chosen to order each pitch set in each fragment according to bandwidth. For each parsed pitch segment we will again establish a proportional relation: all pitches/highest pitch. These proportions will be used as snapshots of seven sonic states in a plane of a three-dimensional array (x, y, z), each state being the sum of all energy weights within one window. We will use them to determine our durations throughout the composition. The durations are of two orders types:

- Macro durations that represent metric time and determine a subliminal pulse illustrated by dynamics. Measures are calculated following the proportions computed from the last segment.

- Local durations consisting of effective duration values from the four instruments. These are distributed according to proportions on either side of the measure bars, creating asymmetrical crescendo-decrescendo pairs.

The main concept of the whole work dealing with pitch/bandwith/dynamic weights is that of symmetry. As we have seen in the example above, we can use it as a compositional element.

Starting from one mode of resonance which was assigned to durations following our proportional translation (see figure 12), we will apply to it a new axis of symmetry where all durations will start from and then continue in asymmetric mirroring, as shown in figure 13.

This was calculated by the `reson-chord` box (see figure 14) and then quantified (figure 15).

[6] Analysis could be thought of as another potential aspect of a sound file, or in other terms, it is an alternative reading/representation of sound.

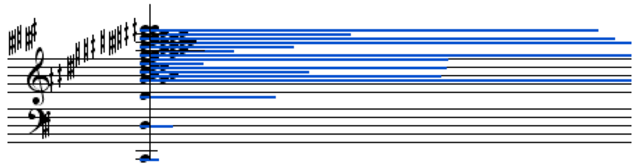

Figure 12. Resonance mode durations.

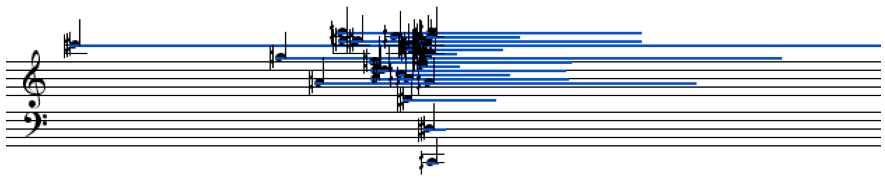

Figure 13. −35 degrees symetrical axe.

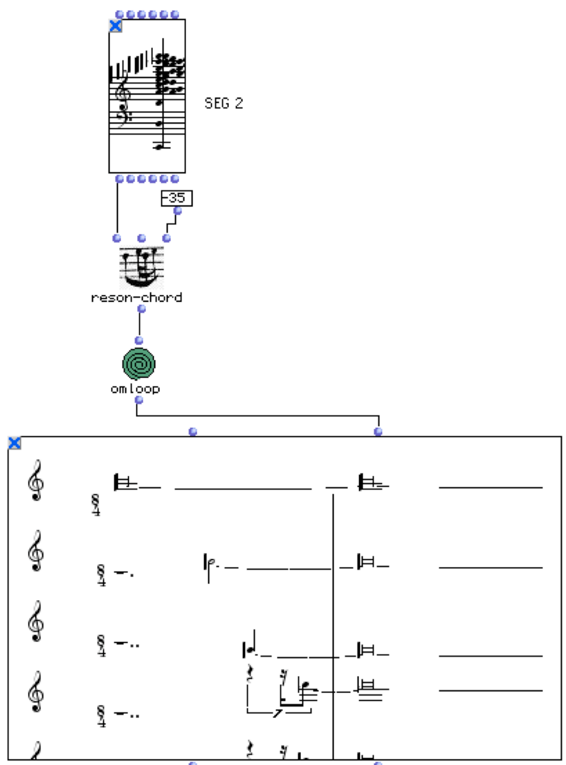

Figure 14. −35 degrees symetrical axe patch.

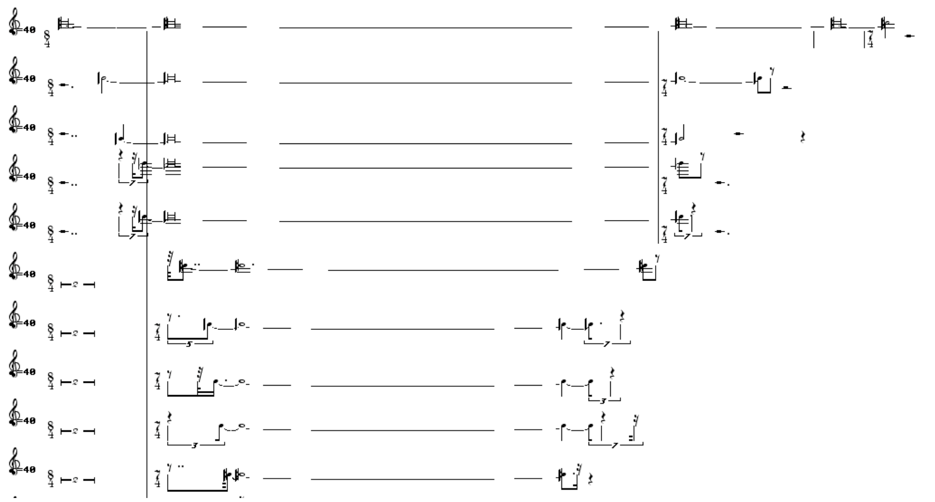

Figure 15. Quantification.

Durations are not the only elements calculated from analysis. Starting from measure 59 (figure 16), pitches are extracted from analysis and distributed over all four instruments following an order based on bandwidth over amplitude generating weight criteria, in descending order of values.

Figure 16. Excerpt from *No one to speak their names (now that they are gone).*

3 Hypercomplex minimalism

3.1 Sound analyis for controlling instrumental gestures

In contrast to the examples we have already seen, where data took the form of 3D information arrays, and which were therefore complex, here we see concrete use of a simpler 2D sound data array.

PTPS analysis is a pitch estimation analysis (pitch and time arrays). When applied to a noise source or inharmonic sound the analyis output tends to yield interesting profiles (figure 17).

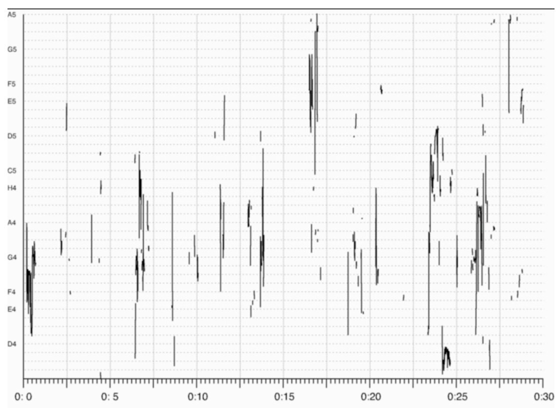

Figure 17. PTPS analysis.

This data will be used after being broken down into n fragments as a means of controlling musical gesture (figure 18).

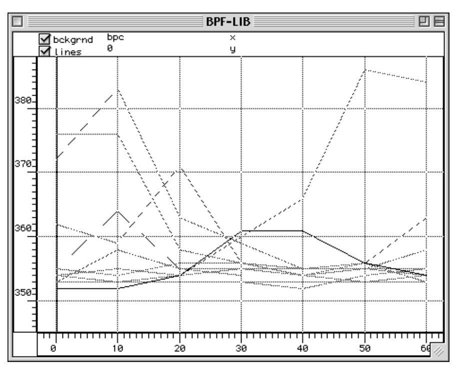

Figure 18. Fragmented analyis in OpenMusic.

The fragments will be considered as potential data for the dynamic control of instrumental gestures (figure 19).

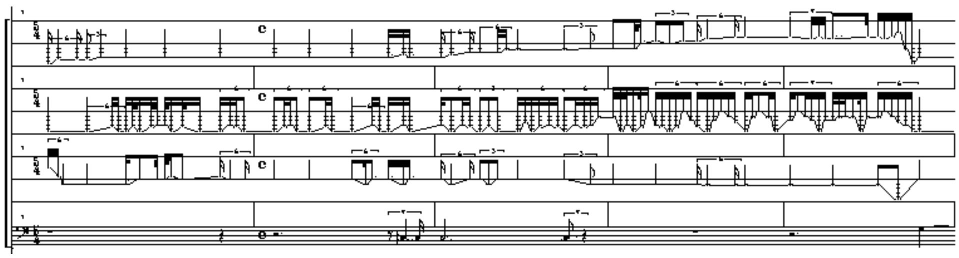

Figure 19. Bow pressure, placement and speed control of the doublebass part in *"In lieblicher Blaue..."* for amplified bass saxophone, amplified doublebass and electronics.

These "potential" fields will be subsequently filtered and assigned according to musical context. As we mentioned above (section 2), the relevance of this technique arises from the fact that all sound sources used for analysis or in the tape part of the piece are taken from samples pre-recorded by the musicians, using special playing modes (multiphonics, breathing, etc.) (see figure 20).

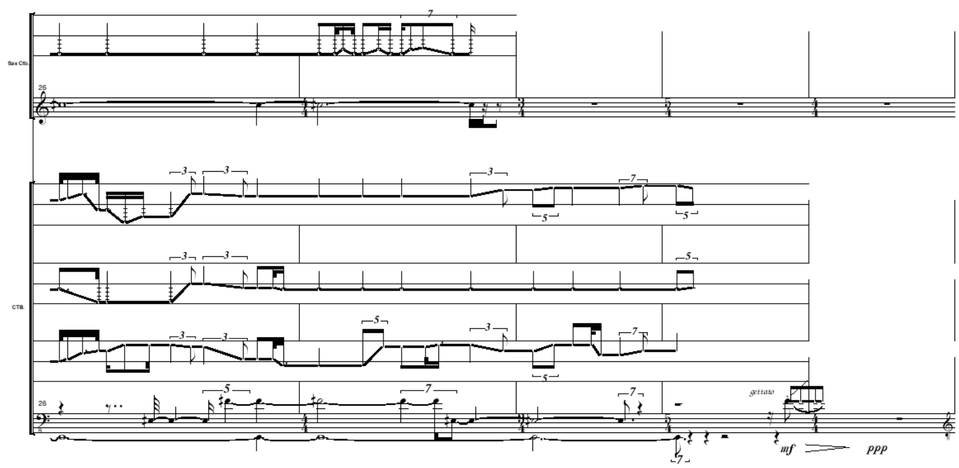

Figure 20. Excerpt from *"In lieblicher Blaue..."* for amplified bass saxophone, amplified doublebass and electronics.

One however must also take into consideration the fact that musical events are a balance between complex gestures in the process of sound production and minimal activity in note and rhythm production, i.e we might distinguish two layers of activity: "traditional" score notation, and control notation.

3.2 *Adagio* for string quartet

Again in this work, a soundfile served as starting point for the whole piece. However, the technique is completely different. Instead of using an external analysis program, all the work was carried out in OpenMusic.

Using OpenMusic's handling of soundfiles, which is limited to playing and representing them in the `sound` object under a time/amplitude curve, typical of all sound editors, it was my intention to use limited and reduced data in order to create a closer affinity with the symbolic mode, keeping in mind the instrumental nature of the string quartet.

I therefore used the `BPC` object and downsampled the amplitude curve in order to have a globally satisfying overview of the amplitude profile (figure 21).

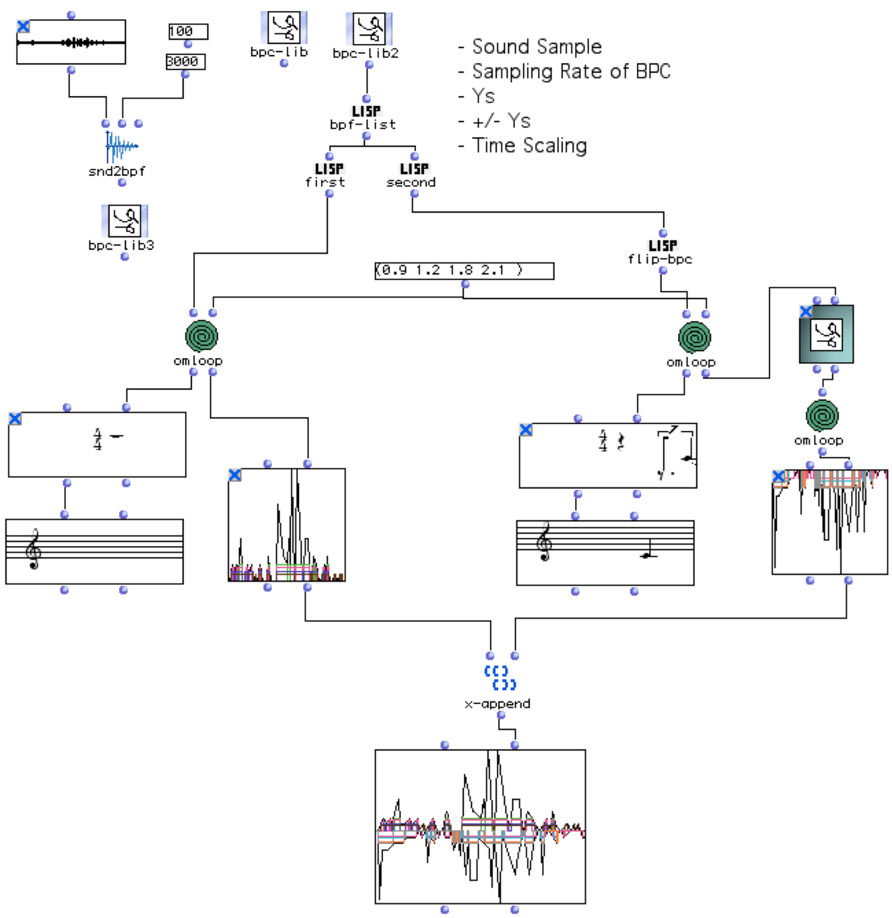

Figure 21. Generating durations from an amplitude enveloppe.

The amplitude having two phases and due to downsampling a more accentuated difference was therefore created between the positive and negative phase creating a double choir polyphony (figure 22).

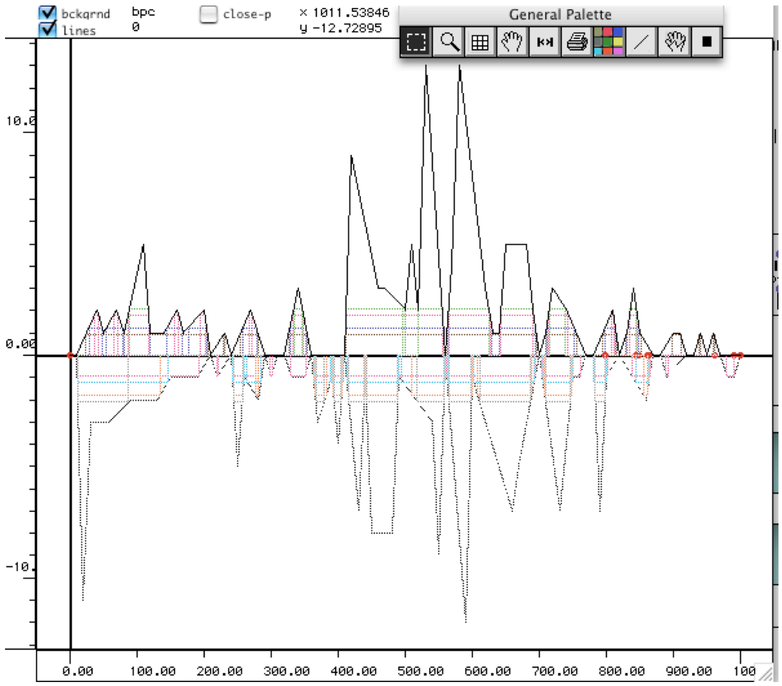

Figure 22. Amplitude curve transformed into double choir polyphony.

I determined four axes intersecting the curves. Duration values were then quantified starting from these segments (see figure 23).

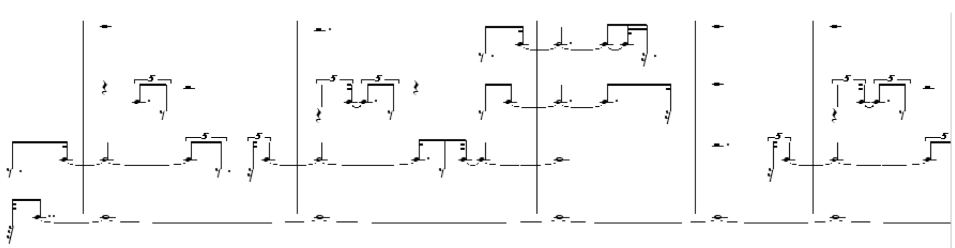

Figure 23. Quantified durations.

In order to verify the result, the two polyphonies were synthesised using Csound and the result was put in OpenMusic's maquette object (figure 24).

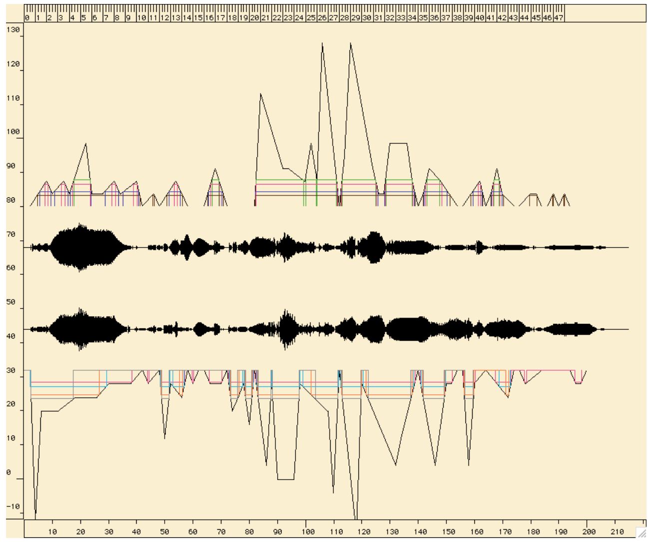

Figure 24. Result displayed in a maquette.

Figure 25 shows the beginning of the *Adagio* for string quartet which was written using the strategy describe above.

Figure 25. Beginning of the *Adagio* for string quartet.

4 Conclusion

In the compositional processes presented here, we can distinguish two fundamental aspects: data and algorithms.

Data in itself can be regarded as a conceptual material. It represents the deterministic drive of "what must be" in a given lapse of time decided by the composer's Deus ex machina.

The algorithms may be seen as the executive part of the composer's idea, also deterministic when the data proposal is added, with dynamic decisional potential that models the propositional data to its own creative role.

These two procedures or techniques are elements of a broad dynamic program, the computational part (analysis and processing) having been carried out with different computer programs such as OpenMusic, or Diphone, and can be considered part of a unique program: the composition itself. It is legitimate nowadays to consider a work of art from the performance and aesthetic viewpoints, but also from a deconstructural angle. I personally adhere to Hegel's thesis [6],[7] siding with his view that art has fulfilled its purpose, and that modern art cannot be understood in the same way as the art that preceded it (from Descartes to Kant). Neither the post-modernist attitude nor techno-classicism will allow the destiny of modern art to be accomplished. A meticulous study of the state of art and of its own medium is necessary, something like the Renaissance. The French composer and philosopher Hugues Dufourt states *"La musique en changeant d'échelle, a changé de langage*[8]*"* [4]. Techniques in composition and sound exploration must be integrated totally not only in the praxis of composition but in its understanding, and better, as a wholly part of composition itself.

[7] "In allen diesen Beziehungen ist und bleibt die Kunst nach der Seite ihrer höchsten Bestimmung für uns ein Vergangenes" (X, 1, p. 16) "In all its relations its supreme destination, Art is and stays to us something that has been" (X, 1, p. 16).

[8] In changing its scale, Music has also changed its language.

Bibliography

[1] Agon C., Haddad K. and Assayag G., "Representation and Rendering of Rhythmic Structures", *WedelMusic*, Darmstadt, IEEE Computer Press, 2002.

[2] Agon C., *OpenMusic : Un langage visuel pour la composition musicale assistée par ordinateur*, PhD thesis, Université de Paris 6, 1998.

[3] Boulez P., *Penser la musique aujourd'hui*, Gallimard, Paris, 1987.

[4] Dufourt H., *L'oeuvre et l'histoire*, Christian Bourgeois Éditeur, 1991.

[5] Deleuze G., *Le Pli - Leibniz et le baroque*, Les éditions de Minuit, 1988.

[6] Hegel F., *Phänomenologie des Geistes*, Hg. von O. Weiss. Leipzig, 1907.

[7] Stockhausen K., "...wie die Zeit vergeht...", *Die Reihe*, n°3, Herbert Eimert ed.,1957.

Karim Haddad

Karim Haddad was born in 1962 in Beirut, Lebanon. He first studied music at the National Conservatory of Beirut, and philosophy and literature at the American University of Beirut. In 1982, he settles in Paris (France). He awards a B.A. in musicology at the Sorbonne University, and follows at the Conservatoire National Supérieur de Musique de Paris studies in harmony, couterpoint, fugue, orchestration, analysis and composition with Edith Lejet (harmony, fugue), Bernard de Crepy (counterpoint), Paul Mefano (orchestration), Jacques Casterede and Alain Louvier (analysis), and Alain Bancquart (composition) where he obtains six rewards and the Diplome Superieur in Composition. He takes part in workshops in composition with Klaus Huber and Emmanuel Nunes, and between 1992 and 1994 he participates in the Ferienkursen fur Musik of Darmstadt where he works with Brian Ferneyhough, and obtains the Stipendienpries 94 in composition.

In 1995, he follows the computer music courses at IRCAM, and becomes member of IRCAM Forum where he contributes in 1999, by writing the *Om2Csound* library for controlling synthesis through OpenMusic environment. He will then write *OpenMusic Reference and Tutorial*. Presently, Karim Haddad works at IRCAM as a technical advisor for the IRCAM Forum.

His works are performed by various ensembles and artists such as the Berlin StaatsOper, l'Itinéraire, 2e2m, Orchestre Philarmonique de Radio-France, Diotima string quartet, ...