Mixing Time Representations in a Programmable Score Editor

Carlos Agon, Jean Bresson
IRCAM - CNRS STMS / Music Representations, Paris, France

***Abstract*— This article presents a new type of document developed in the computer-aided composition environment OpenMusic. The *sheet* extends the compositional tools of this environment by putting forward the concepts of score and music notation. Two principal questions are dealt with: the integration of musical objects corresponding to heterogeneous time systems (e.g. traditionnal music notation representations and linear time representations), and the integration of programs and functional descriptions with the notation aspects within the score framework.**

I. Introduction

Current computer music systems must deal with the integration and linking of heterogeneous objects and information related to musical and multimedia data structures and content, and with their management in complex processes (calculus, storage, codification, visualization, edition, etc.) Particularly, the objects that can interact in composition processes may be related to different temporal specification paradigms and representations, which impose a specific consideration about the integration of heterogeneous time systems.

Contemporary and electroacoustic music are particularly concerned with this issue, since composers often deal with time durations, or even directly with the continuous aspects of sound structures, instead of (or in relation with) traditional pulsed time representations (see Boulez's *temps lisse* and *temps strié* [5]). It is indeed frequent to meet situations where musical objects (chords, voices, etc.) are parts of scores where different other types of data, dedicated to be interpreted by performers or by sound synthesis or processing programs, are also present. Composers must generally carry out such scores' edition manually using graphical tools, after the composition process is over. No interaction between these heterogeneous components is therefore possible in the score framework during the composition process.

Nonetheless, the score is still considered as the principal, or at least as a preferred support for music writing. Even when music is not meant to be read or interpreted (e.g. in the case of electroacoustic music), this document can have a great importance as a formal support and a privileged place for for musicians to write and read music, i.e. to think and understand it [4]. It is therefore the place where the aforementioned issues should particularly be considered.

Examples of problems commonly met are the cross representation and manipulation of rhythms and proportional duration representations (distinct objects and structures must generally be used in order to work independently on one or the other representation), or the handling of static musical structures together with dynamic links or functional relations.

The computer-aided composition environment OpenMusic [1] [3] addresses such problems and proposes different modalities for these structural and/or temporal integrations. In this article we present a new score editor created in this environment, called *sheet*. The *sheet* editor deals with the heterogeneous objects and time systems integration and with the visual representation and edition of the data and programs in music composition. The objective of this work is to gather in a document the characteristics of a score as a readable document (which implies providing a coherent temporal representation) and the (visual) programming features of a computer-aided composition environment.

II. SHEET: General Presentation

The *sheet* is an OpenMusic object which can be seen both as a document and as a musical object.

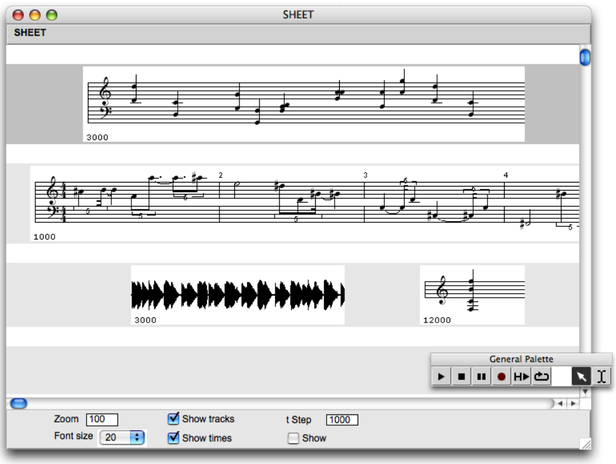

Fig. 1. A *sheet* editor. Each voice contains one or various objects starting at different times.

As a document, it keeps following objectives of the computer-aided composition environment concerning the integration of musical material and programs in the compositional framework. This document is centered on the concept of score, in the traditional sense. In addition to the features of the former OpenMusic score representations [2], however, the *sheet* aims at mixing traditional musical notation with the new types of objects and structures possibly involved in contemporary music.

The *sheet* can gather all kinds of objects in a common temporal representation. It is composed with a variable number of voices, each containing one or various sequenced objects. As such, it is also considered a musical container in the composition environment. In principle, any type of object coming from the OpenMusic composition framework can thus be inserted in a *sheet*, provided it has a duration: chords and sequences, voices, MIDI files, but also curves and envelopes, sound files, etc. (Figure 1 shows an example of a *sheet* containing various types of musical objects.)

These different voices can be created in programs defined in the OpenMusic patches and gathered/inserted in a *sheet* manually (drag and drop) or directly in these patches. Figure 2 shows a *sheet* built in a patch from a list of musical objects.

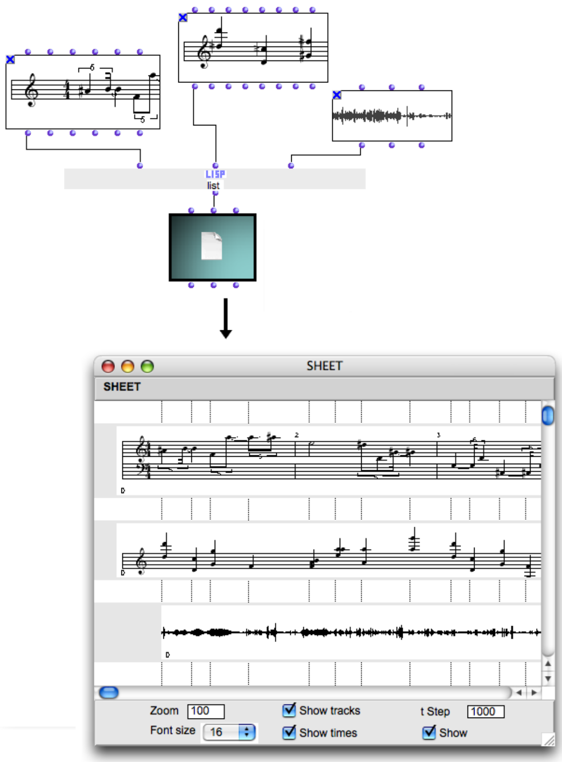

Fig. 2. Building a sheet from within an OpenMusic patch editor.

The basic functionalities in the *sheet* editor are thus the creation/removal of voices and the additon/removal of musical objects to/from these voices. The voices display can then be edited (size, positions, spacing, show/hide background, etc.) in order to reach a desired graphical disposition of the score.

Each one of the inner objects is accessible and editable using an independent editor (score editor, BPF editor, sound file editor, etc.)

III. Representation of Heterogeneous Time Systems

One of the main problems dealt with by the *sheet* is the consistent common graphical representation of the different objects it contains.

A. Non-Linearity in the Time Representation

The music notation is not linear regarding the time representation: there is no linear relation between time (durations) and the graphical space occupied by the symbols in the score. In Figure 3, a temporal grid with a step of 1 second is displayed on the score. It is evident that the space occupied by one second constantly varies along this score.

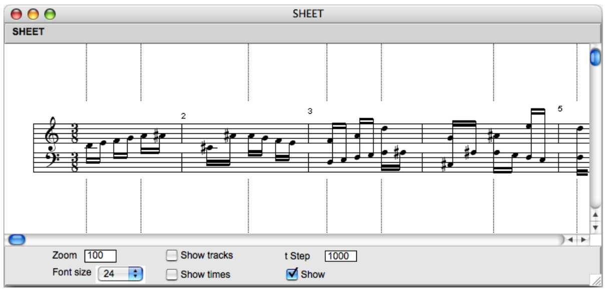

Fig. 3. A voice in a *sheet* editor with a regular temporal grid display.

A typical example of this non-linearity is the display of the score elements that have certain graphical dimensions but do not take any time (e.g. the keys, measure bars, metric information, alterations, etc.)

This non-linearity is a problem when various objects need to be represented together, since the simultaneous events should be exactly at the same position in the horizontal (temporal) axis. In Figure 4, a canon is represented, that is, two identical voices are present, with the second one shifted (by 1 second, in this case): they are identical in their content but the graphical representation is distorted in order to respect the temporal alignment of the musical events. For example, the second note of voice 2 uses almost twice the space of the second note on voice 1: some space was required there for displaying the first bar in voice 1.

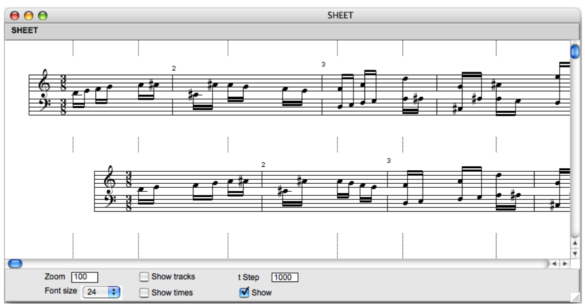

Fig. 4. Two parallel voices in a *sheet* editor: time/space distortion.

B. Temporal / Graphical Alignment

The consistency of the temporal representation is ensured in the *sheet* editor by an algorithm that calculates a function relating time to space in the score, given the objects currently present in it.

All the possible landmarks in the musical objects are considered, in order to determine ratios linking time (the duration between the successive landmarks) with space (the dimension of the corresponding score graphical components). Gathering and mixing these time/space values for all the objects present in the score allow to get a global discrete function $x = f[t]$ in which all the score elements are given a graphical position accordingly to the possible non-linear constraints of their containing object (as in the case of figure 3) but also to that of all the other objects of the score (figure 4).

Applying this function to the representation thus distorts the relation between time and space for every objects in the score (including those that could commonly be represented in a linear way). All of them are graphically stretched or compressed in order to respect the distortion brought on by the "non-linear" notation symbols. (As it occurs in music notation editors, the playing cursor of the *sheet* editor will also adapt its scanning speed according to the time/space function.)

This method will therefore allow the common representation of objects corresponding to heterogeneous time systems. Determining landmarks in an object brings non-linear constraints on the global representation, or simply allows to create the corresponding point in the time/space function in order to adapt its representation to that of the other objects.

C. Rhythmic Notation and Proportional Notation

Figure 5 shows an example of a *sheet* editor where the two voices now correspond to two different time systems: a voice is in traditional music notation, like in the previous examples, and the second one is a MIDI file where events and durations are specified in milliseconds (which is sometimes called "proportional", or "linear" time representation).

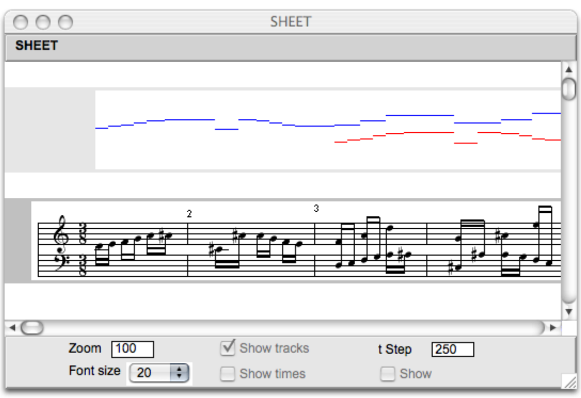

Fig. 5. Mixing linear and non linear time systems : a MIDI track (linear) and a voice in music notation (non-linear).

The beginnings and endings of the MIDI events are considered as landmarks, so that the onsets and durations of the MIDI events are graphically stretched following the uper voice's score notation constraints in order to coincide with its notes and symbols. Two MIDI notes of the same duration (in this example, they actually all have the same duration) will thus not necessarily have the same graphical size anymore.

In OpenMusic, the *chord-seq* is another "proportional time" object represented in musical notation: although it resembles a score representation, each chord or note is specified by an onset given in milliseconds, without consideration for any other meter or pulse unit. However, in contrary to the case of the MIDI file (figure 5), the notes in the *chord-seq* carry non-linear graphical display constraints, since they are drawn in the traditional way on a score system (note heads, possible alterations, etc.) The *sheet* editor, in this case, will thus also allow for the representation of *chord-seqs* together with the other types of objects (and in particular with rhythmic notation, as shows figure 6).

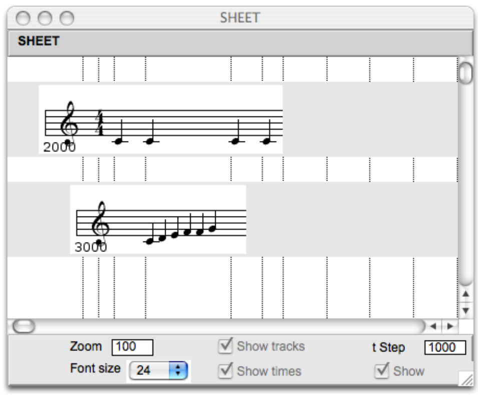

Fig. 6. Mixing rhythmic and linear score notations: each note in the upper (rhythmic notation) voice lasts 1s. The 2^{nd} second is stretched in the representation in order to make the *chord-seq* notes fit between the second and third notes of this voice.

D. "Continuous" Objects

In addition to the previous types of temporal systems, there exist some objects which we call "continuous", in the sense that they do not contain events (like notes). These objects include curves or breakpoint functions, as well as sound signals.

In these objects, two important moments are known in principle: the beginning and the end. These moments allow the positioning of the objects and the deduction of their duration in the temporal representation of the editor. In many cases, however, additional points of interest should be considered between these two moments: distinction of sub-segments, highlighting of particular events not explicitly represented in the objects, etc. Making these moments participate to the time/space function would indeed allow to correctly position them in the global time representation, instead of merely make the continuous object representation depend on the other (external) events of the score.

Hence, landmarks can be defined (or automatically determined) inside these objects in order to refine their graphical representation by applying the possible graphical distortions independently on each subsequently deduced segment.

In the curves and functions, the inflection points (or control points) can be used as landmarks. In the case of sound objects, we use markers manually defined along the time axis.

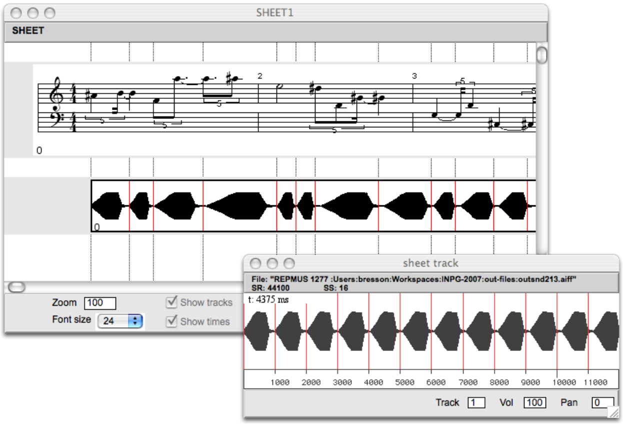

Fig. 7. Mixing temporal systems in the *sheet* editor: a traditional voice and a sound file. The markers in the sound allow to align highlighted moments to the score representation.

Figure 7, shows an audio file, in which some regular markers were placed at each second, beside a voice expressed in traditional music notation. The beginning of the audio track corresponds to the real beginning of the voice (i.e. at the time of the first note of this voice), and the segments between the moments highlighted in the audio file with the markers (in this case, of one second each) are stretched or compressed in order to respect the time spacing of the score.

IV. Programs in the Score

The second main feature of the *sheet* editor are the programming possibilities provided in the score, which allow one to set functional relations between the objects. Indeed, we believe (this is one of the principal bases behind the development of OpenMusic) that programs should be considered integral part of the scores representations in computer-created music.

By switching the *sheet* editor to the "patch" mode, the user is provided with the possibility of bringing to light another part of the musical semantics of the score, concerning the functional and generative aspects (i.e. the way the objects are created and/or linked between one another.) In this "patch" mode, the editor comes closer to the usual patches of OpenMusic: a visual program editor in which each object is displayed with an input and an output and can be connected to functional call boxes and/or to the other objects of the score.

In Figure 8, for instance, two voices are connected through a program that makes one the inversion of the other. The possibility to evaluate the objects (or the whole score) allows to (re-)built the contents of the score according to their possible dependencies. In this case, whatever the modification done on the first voice, evaluating the editor will make the second one its inversion.

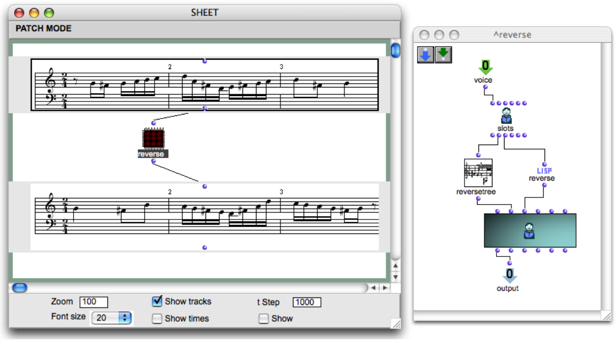

Fig. 8. Functional relations between the components of the *sheet* by programming in the editor ("patch" mode).

If we consider again the example of Figure 7, one might suppose or wish that the electronic part (the sound) would be related to the upper (assumed instrumental) voice. Figure 9 is a related example where a sound synthesis process is carried out in the score editor: the sound file is computed starting from the data coming from the notes of the first voice [6].

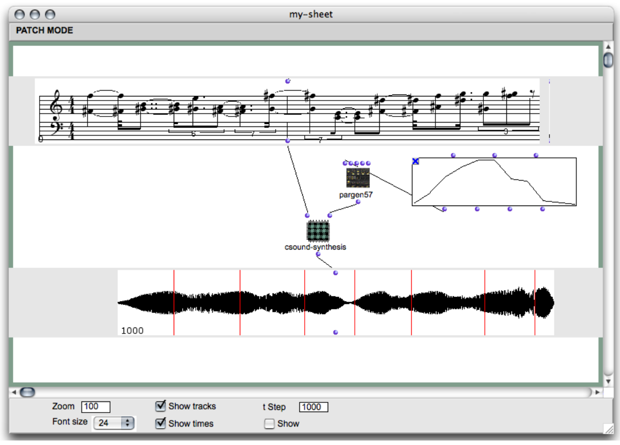

Fig. 9. A sound synthesis process in the *sheet* editor ("patch" mode). The *csound-synthesis* box is an abstraction of a program that carries out a sound synthesis starting from a voice and some additional data (an amplitude envelope).

The representation can thus be enriched with the different objects and programs taking part in the musical and/or sound synthesis processes (which, from the reader or music analyst point of view, might also allow a better understanding of the score.) By switching back to the "score" mode, the programming components are hidden for a clearer display.

V. Conclusion

The *sheet* was introduced as a new document and as a musical object in the OpenMusic computer-aided composition environment. Heterogeneous time representations are dealt with as well as programming issues, which we both believe to be of a major importance in computer composition systems.

The *sheet* extends the other documents available in this environment (the patches and the maquettes) and may be used for various purposes in complement with them, particularly when there is a special interest in notation. All these documents can interact in compositional processes: patches can be embedded in the *sheet* in the programming mode, and maquettes can be embedded as well, either as programming or as musical objects. Symmetrically, the *sheet* can itself be algorithmically constructed in the patch editors, this time as a musical object. These multiple embedding possibilities of the documents one in another thus allow composers to combine freely the various abstractions and structural levels used in their composition framework.

The *sheet* editor may be particularly relevant as a support for creating scores integrating instrumental parts (written in symbolic musical notation) with sound signals or other instructions related to an electronic part of a piece. Thereby, it might provides a convenient way for editing and notating different kinds of contemporary music pieces.

Future works should improve the temporal relations, especially by providing means to set temporal synchronization constraints between the objects. An object could indeed be positioned in the time axis by synchronizing it or one of its internal components or landmarks with a component or landmark of an object from another voice.

Another important work remaining to be done concerns the pagination of the score editors, in order to allow for the display and printing of large scores and musical works.

References

[1] Agon, C. *OpenMusic : un langage de programmation visuelle pour la composition musicale*, Ph. D. Thesis, Université Pierre et Marie Curie, Paris 6, 1998.

[2] Agon, C. and Assayag G. Programmation visuelle et éditeurs musicaux pour la composition assistée par ordinateur *14ème Conférence Francophone sur l'Interaction Homme-Machine IHM'02*, Poitiers, France, 2002.

[3] Assayag, G., Agon, C., Fineberg, J. and Hanappe, P. An Object Oriented Visual Environment for Musical Composition, *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, 1997.

[4] Assayag, G. Computer Assisted Composition Today, *1^{st} Symposium on Music and Computers*, Corfu, Greece, 1997.

[5] Boulez, P. *Penser la musique aujourd'hui*, Gonthier Ed., 1964.

[6] Bresson, J., Stroppa, M. and Agon, C. "Symbolic Control of Sound Synthesis in Computer-Assisted Composition". *Proceedings of the International Computer Music Conference*, Barcelona, Spain, 2005.