Towards Digital "Musical Actions"

Alain Bonardi  
Paris IV Sorbonne University  
5, impasse du Débarcadère  
78000 Versailles, France  
+33 1 39.51.74.13  
alain.bonardi@wanadoo.fr

Francis Rousseaux  
Reims University  
20, rue de Condé  
75006 Paris, France  
+33 1 46.33.60.03  
Francis.Rousseaux@lip6.fr

ABSTRACT
In this paper we present our research about a new genre of lyric works, called virtual interactive operas. Part of its results are used in the composition of an interactive opera called "Virtualis". To achieve interesting hypermedia interaction in a musical context, we show that it is necessary to use a description language and we give some specifications of the one used. Using this language enables the creation of graphical metaphors of structural elements of music, with the help in the future of the EMMA metaphor builder system. We then describe the ALMA authoring system we have been developing to implement hypermedia musical compositions.

Keywords
Virtual interactive operas, hypermedia authoring system

1. WHAT ARE VIRTUAL OPERAS?
We define as a virtual opera any opera implemented on a personal computer [4][5], enabling some interactivity with its audience. This new genre is not based on the simple transposition of existing operas in the framework of new computing technologies; it takes into account the shiftings of uses and meanings induced by multimedia computing, assuming they could arouse new writings, including the lyric field.

Interactivity is based on free choices among different paths. Only open forms [8] that have specially been designed for listeners can enable it. This specification leads us to give up the classical narrative "aristotelician" mode [14]. This implies two possibilities of formal and temporal articulation:

- either consider the hypertext model and possibilities of automatic text generation [2] to design an opera where hypermedia components strongly depend on text; this is the case in Jean-Pierre Balpe's Barbe-Bleue project,

- or take into account the specificities of computer-aided writing to design a new kind of "musical action" as Luciano Berio [3] defined it: "Between a musical action and an opera, there are substantial differences. Opera is based on a 'aristotelician-like' narrative mode, which tends to have priority on musical development. On the contrary, in a musical action, the musical process rules the story".

We have chosen this very direction for our research and our Virtualis project.

One of the consequences of this choice is that music is no longer for us a simple resource accompanying an interactive plot. Music entities become kind of characters, that can be freely accessed through pointers. This new genre requires new adequate composing tools to be used in the framework of such classical multimedia development tools as Director.

In this paper we show the necessity for a description language dealing with music to complement the Lingo language. This also enables a strict separation between formal descriptions and resource files (for instance sound files) [1]. We then describe the ALMA system we have been developing for that purpose.

2. A DESCRIPTION LANGUAGE
In our approach music is no longer a simple passive resource; it is therefore necessary to define categories that can be handled by the composer. On the other hand, in the framework of virtual operas, we need fine interactions between music and other components such as graphics, going further than basic temporal synchronization.

For instance, we would like to generate 3D visualizations of music pieces that would be kind of sets of our opera. A language to describe music should be very useful to complement the Lingo language used to specify graphics and interactions in Director. It should use plain text, be directly readable and independent of any technical platform.

The main specification of this language would be its ability to annotate music. Let us consider the western musical notation system as it has been formalized since the second half of the eighteenth
century. This system includes two separate but complementary spaces:

- the first one is a dual space based on the category of note, mixing two subcategories:
  - the pitch which cannot be thought independently from the vertical axis used to represent it. Its notation is therefore proportional: the higher the frequency is, the higher it is placed on the staff, the higher we think of it in physical terms.
  - the duration based on algebraic relationships, more precisely on divisibility in Western music: a whole equals two halves, etc.

- the second space includes the varied prescriptive notations: they are sometimes precise indications such as the tempo; sometimes contextual indications (for instance “in tempo di marcia” which means “at the tempo of a march”, a march being a very precisely encoded baroque dance); sometimes notations close to rhetoric such as slurs or accents. It is possible to classify them in two groups: on one hand, causal notations (“ponticello” for strings indicates the position of the bow, and does not describe the sound itself); on the other hand, effect notations (for French horns, the “+” notation indication tells that the sound has to be brassy).

How do computer-aided music representations deal with these two levels? The most commonly used it the MIDI format, which takes into account the first space, that is the pitch and duration space and restricts the second one to causal notations with quantified parameters. On the other hand, it has no possibility of effect notation, even basic ones linked to pitches; thus, it is impossible to define a musical phrase. Generally speaking, MIDI cannot handle any aggregation, either temporal or harmonic.

We must therefore use a language able to smoothly annotate music. We have specified an extended version of GUIDO [10] for our needs, which is based on the highlighting of music structures [7] on various levels:

- the first level has to do with the temporal links within the piece. As in sequential programming languages, three kinds of linking can be used:
  - sequences, that is various notes or events or groups of notes or groups of events played one after another. In GUIDO, brackets are used to encode sequences. For instance, [C1*1/4 D E C] plays four quarters starting from the central C on the keyboard.
  - loops, that are encoded with \RepeatBegin and \RepeatEnd tags.
  - test conditions, which happen for instance in open forms written for orchestra when the conductor decides to play or not to play such or such section. When describing the ALMA system, we will show how these tests are implemented through links.

- on the second level, we deal with aggregations:
  - either temporal aggregations: the \Slur () tag indicates a musical phrase.
  - or harmonic aggregations: braces indicate a chord. For instance {C1*1/4 E G} encodes a perfect triad lasting one quarter.

For instance, let us consider the very beginning of the Diabelli Variations opus 120 by Beethoven:

![Image of music notation]

The GUIDO translation of this extract is:

{[staff<1 \slur(_/4 _ *7/32 D2/32 C/8 B1/8 C2/4) \slur{C1 E G} {C1 E G} {C1 E G} {C1 E G} {C1 E G} {C1 E G} {C1 E G} {C1 E G}} [staff<2> _*3/4 C0/4 \slur(G-1 C0) \slur(G-1 C0) \slur(E-1/8 F G E C/4 _)]}

3. GRAPHICAL METAPHORS OF MUSIC

This description language enables the creation of mappings between structural elements of music and graphical elements displayed, that is metaphors, which can have three different roles of mediation between music and listeners:

- the first role is to make explicit musical structures and processes handled by composers and analysts; metaphors are then used as translators.

- the second approach is interested in the listener’s point of view, notably using results from psycho acoustics: in a given context, what does the listener perceive, which categories are relevant for him? That was the basic idea of François Pachet and Olivier Delerue [11] when developing their MIDI Spatializer named “Midispace” which enables
users to move instruments on a stage, the program taking into account consistency constraints simulating the action of a sound engineer.

- the third role consists in arousing a kind of convergence between music and graphism, without making it necessarily explicit. The listener does not then know how to designate the musical phenomenon he has perceived, but he/she is able to associate it with geometric or color configurations.

We have implemented simple graphical metaphors of music. For instance, our “Musical Tunnels” interface enables the creation of corridors whose shapes and colors are computed from the analysis of musical properties. This program for Macintosh is written in C and uses QuickDraw3D libraries. The next three images show the result with the Diabelli Variations by Beethoven.

figures 1, 2, 3 : different views of generated tunnels

Gates correspond to the beginning of musical phrases and when the listener goes into the corridor, the music is triggered. Then the listener wanders in the tunnel at the pace of music. Metaphors have an emotional role here, since they underlie musical aspects that a non-musician can perceive without knowing what it precisely is. But this synesthesia enables the listener to visually memorize striking aspects, and that was emphasized by almost all users of the system.

We also studied a way to visualize polyphonies. In the following interface, the user can create graphical metaphors of any GUIDO file and choose to dispatch chords and voices to different slides. We show the results with a piano solo extracted from the Piano Concerto K.488 by Mozart. The left-hand part is dispatched into two slides, whereas the upper slide represents the right-hand part.

figure 4, 5 : different views of generated slides

Our next works will deal with the generalization of metaphors (using works about computer program metaphors [13]), that is designing them as sets of entities with the help of a computer system, with static aspects such as their size (which corresponds to the value of a parameter) or their relative positions in space (which help the user get the situation) and dynamic aspects such as temporalities or exchanges between entities. After defining these entities and their relationships, the user will be able to select the musical aspects (actually encoded according to GUIDO syntax) corresponding to each entity, and the system will check if the set of metaphors is consistent or not. This system will be implemented in C and generate QuickDraw3D MetaFiles that are then imported in Director. We hope that this future system, called EMMA, will provide new possibilities of metaphor creation to be used in musical multimedia applications.

4. THE ALMA ENVIRONMENT
Handling music in computer-aided open forms requires a specific environment for creation. It first implies that music cannot be considered any longer as a resource stored in files. Moreover, classical authoring environments as Director do not enable the composer to have a clear apprehension of the open form he wants to create since they were not designed for music purposes.

We have therefore started developing our own authoring system, named ALMA. This is a hierarchical object-oriented system where logical descriptions of music or stage indications are separated from physical files to be played, either music files (audio or MIDI files) or graphics files (created with the EMMA metaphoring environment). The ALMA environment was developed in Lingo with Director.

The ALMA system includes different modules:
• two annotation modules; the first one enables to add such descriptions as slurs or harmonic data to MIDI files; the second one is used to annotate audio files thanks to a corresponding MIDI file that was previously annotated with the first module.

• a COMPOSER module to edit and put together entities and links. Entities can be edited through four windows: a first one to edit texts written with the description language; a second window for the score; a third to watch the result of the graphical metaphor chosen for this very entity; a last one to manage physical resources (sound files, 3D generation modules).

• a PLAYER module that performs the interactive score prepared with the COMPOSER module.

5. CONCLUSION
We have shown the necessity of a description language in a context of hypermedia creation, and given general specifications of the one chosen in the ALMA system: it is a language close to GUIDO, with possibilities of links and aggregations. We are planning to develop a metaphor builder environment called EMMA. Working with both ALMA and EMMA environments should arouse new approaches to musical multimedia creation.

REFERENCES
1. AUFFRET, Gwendal, BACHIMONT, Bruno, CARRIVE, Jean, CHEVET Olivier, DECHILLY, Thomas, RONFARD, Rémi (1999), Audiovisual-based Hypermedia Authoring: using structured representations for efficient access to AV documents, Proceedings of the 10th ACM Hypertext’99 Conference
6. BOUCCURECHLIEV, André (1993), Le langage musical, Paris, Fayard
7. CHEMILLIER, Marc, PACHET, François (1998), Analyse de structures, in Recherches et applications en informatique musicale, Paris, Hermes