NeXTSTEP Graphical Interfaces to Control Sound Processing and Spatialization Instruments

Todor Todoroff - todor@musique.fpms.ac.be
Caroline Traube - caroline@musique.fpms.ac.be
Jean-Marc Ledent - jml@musique.fpms.ac.be
Laboratoire d'Informatique Musicale, Faculté Polytechnique de Mons, Belgium

Abstract

This paper presents the new graphical user interfaces we have developed to help computer music composers to interact with the computer. These interfaces allow to control sound instruments both in the FTS Client/Server architecture and through MIDI. We present more in detail a movement generator interface that is specially useful for sound spatialization. We also describe how we integrate these graphical interfaces in a more global environment from which the user may both launch applications and establish connections between them.

1 Introduction

The graphical interfaces are programmed on the Ircam Musical Workstation (computer NeXT + one to three ISPW-16 cards). Its operating system is the object-oriented and graphical environment called NeXTSTEP in which we realize graphical interfaces thanks to development tools such as Interface Builder and Project Builder. The real-time sound processing environment called FTS, running on the ISPW-16 card, is able to communicate with the NeXTSTEP applications through its client-server architecture or through MIDI.

Actually, each application is able to send and receive MIDI messages through a MIDI interface connected to the NeXT serial interface. In that way, our NeXTSTEP graphical interfaces can communicate with any MIDI device (keyboards, faders boxes, synthesizers, samplers,...) as well as with old MAX 0.26 patches running on a ISPW-16 card. Connections between the computer and different MIDI devices are illustrated on Figure 1.

We are planning to port the graphical interfaces on the future Macintosh OS, called Rhapsody. It will be easy thanks to the integration of the OpenStep concepts and tools in the future operating system. This is the new strategy of Apple since the buying of NeXT Software and the return of Steve Jobs into the company.

We are developing these software tools for computer music composers within the framework of a project financed by the Région Wallonne in Belgium.

2 The Trajectories Generator

2.1 Generalities

This interface is developed under real time object-oriented concepts. It constitutes a graphical tool to define threedimensional trajectories, independently of the spatialization system. As shown on Figure 2, the composer can choose predefined trajectories (simple shapes like circle, square, spiral, ...) and launch them by simply clicking on the corresponding buttons or define his own 3D trajectories with the mouse or with a data glove.

The 3D trajectory is defined with the mouse (or other 2D controllers) in two steps. The user first draws the horizontal projection of the trajectory (left view, in the X-Y plan) then he defines the z function, the altitude curve thanks to the table editor (upper view). The view on the right shows the resulting movement in space.

The trajectory is entered in only one step with the data glove.

2.2 Edition of the trajectories

From one particular trajectory, several other trajectories forming a class can be derived thanks to different

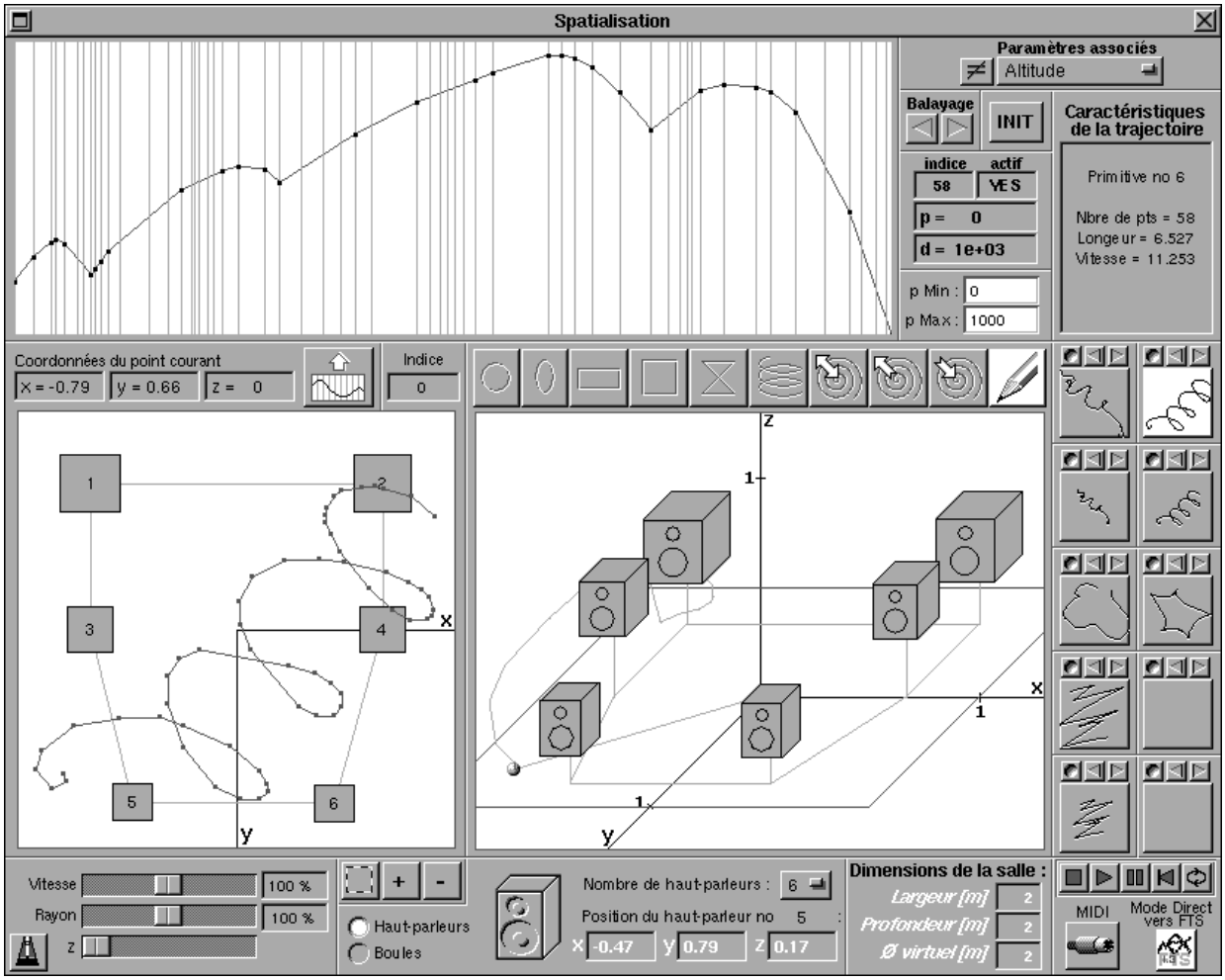

Figure 2: MoveInSpace, a trajectories generator

transformation methods : translation, scaling, speed modification, moving, adding or removing points, ... These trajectories are saved by pressing the record button and are played back by pressing the forward or backward play buttons. During the saving process, an icon representing the trajectory is created. A set of classical commands buttons - Stop, Play, Pause, Return To Zero, Loop - is also available.

2.3 Correlation between trajectory and sound morphological parameters

It is moreover possible to correlate the spatial movement with parameters of the sound (e.g. : frequency, amplitude, filter parameter, ...). The user may use the upper view to define other functions, in the same way he defines the altitude curve. The composer can explore the resources of spatio-morphological transformations, correlating sound spatialization to other sound processing, like pitch shifting, dynamic filtering, ...

As a matter of fact, MoveInSpace becomes a general multi-purpose controller. It allows to simultaneously control the dynamic evolution of a large set of parameters and can be useful to pilot sound processing programs that require the definition of many parameters.

Since the application is able to send and receive MIDI messages, a keyboard can be used to launch the trajectories. A special interface has been designed to associate a key with a trajectory number. Furthermore MoveInSpace could efficiently assist the composer to visualize and modify the parameters of a MIDI sequencer, using its SMPTE and/or MTC synchronization facilities.

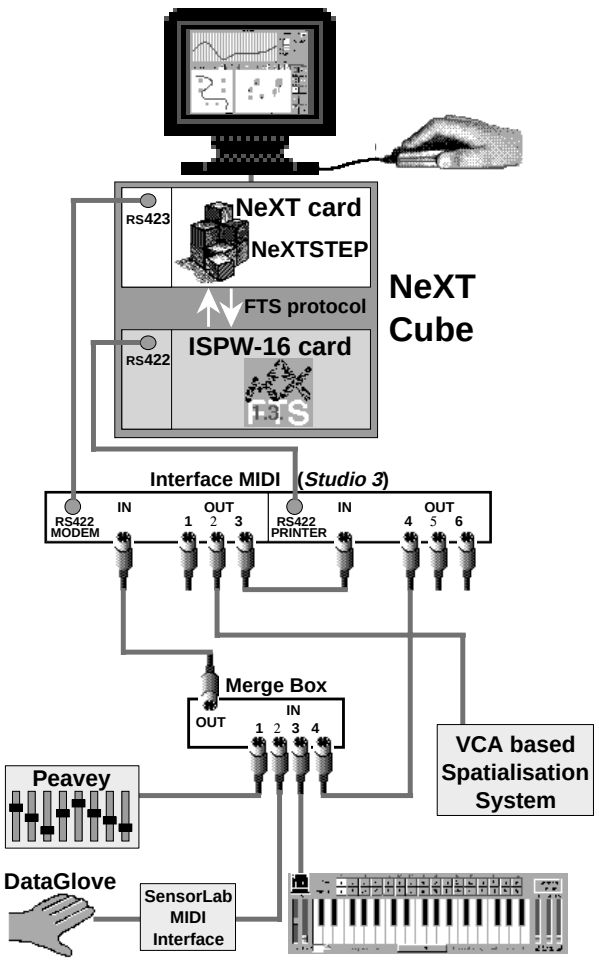

Figure 1: Connections between the computer and other devices

2.4 Room and Loudspeakers Specifications

Different parameters concerning the room where the sound is spatialized are easily specified and visually controlled by the user, such as the real and the virtual dimensions of the room, the number, the position and the size of the loudspeakers.

2.5 Interpolation

We also introduced the interpolation concepts in this interface (see Figure 3). The user may place and size interpolation spheres in space. Each sphere represents a set of parameters and a resulting set is calculated by interpolation, depending on the position of the point moving in this space. Actually, this is the extension of the application Interpolation 2D (see Figure 4) [2] [3] [4].

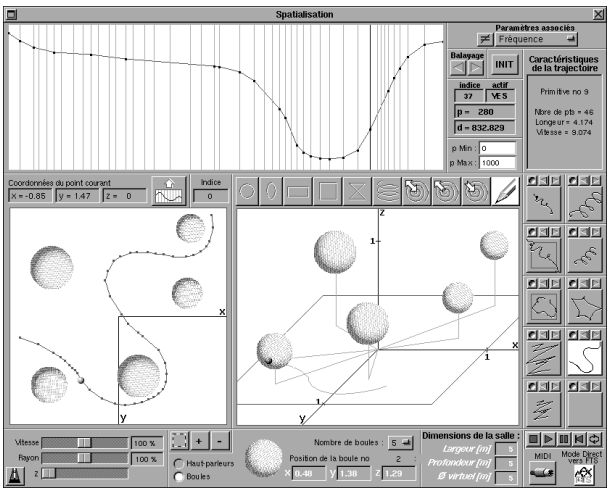

Figure 3: Interpolation Spheres

2.6 Spatialization systems

We are testing this tool on different spatialization systems : on an acousmonium controlled by VCA (through MIDI), on a quadraphonic spatialization system and on the Ircam's Spatialisateur.

We will use this interface to spatialize the music for an open air theatre play this summer. It will be done by MIDI controlling two Yamaha 03D Mixing Desks configured as matrices using the four Aux. sends and the four Subgroups.

3 The Interfaces Manager

The final goal of developing a set of GUIs is to constitute a palette of multiple-purpose and dedicated interfaces from which the composer can choose the ones that best suit the sound processing tools he wants to use. This palette can be seen as a toolbox integrating each single interface's functionality, allowing the composer to create his personal environment by combining the interfaces respective features in powerful and rich interactions, giving free play to his imagination and creativity. Therefore we developed an Interfaces Manager application, providing the composer with a tool to define interactions between the applications he is using.

The Interfaces Manager application allows to launch several instances of the same application,

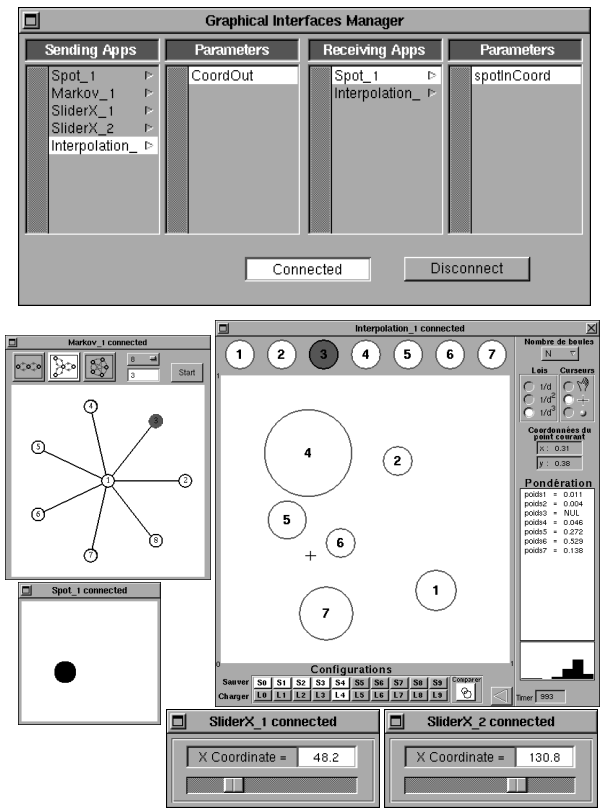

Figure 4: The Interfaces Manager

according to the composer's needs, and to set up asynchronous connections between applications parameters. The connections are set up and closed using a window presenting two two-columns browsers (see Figure 4): on the left side, the browser for sending interfaces and their output parameters and, on the right side, the browser for receiving interfaces and their input parameters. The user can also select scaling and offset factors to customize the conversion of an output value to the input parameter value.

Example of connections between two applications: the composer launches a random sequencing application based on Markov chains (see Figure 4), he then launches the MoveInSpace application and finally uses the Interfaces Manager to link each state of the Markov application with one particular trajectory. Each time the Markov sequencer reaches a new state, it triggers the corresponding trajectory.

Note that the Interfaces Manager as well as the integrated graphical interfaces are independent executable softwares. This structure cares for maximum modularity and flexibility by minimizing the interdependencie between all the collaborating applications.

4 Conclusions

Since a clear and intuitive graphical user interface determines the efficiency of the creative work, we dedicate an important part of our work to the development of graphical user interfaces in order to ease the interaction between the computer and the artist, the composer.

We integrated our graphical interfaces in an environment that both permits to launch them easily and to establish connections between them, as well as with the sound processing equipment. This palette proposes, among others, the MoveInSpace application that constitutes an efficient graphical tool to easily control spatialization instruments or, in general, any sound processing program that requires the simultaneous definition of many parameters.

For the moment, the interfaces presented here are available on NeXT computer and on stations running the OpenStep OS (Sun, PC, ...). In the future, they will be available on Macintosh computers thanks to the integration of the OpenStep concepts and tools into the Mac OS Rhapsody.

References

[1] Déchelle, F. and M. De Cecco 1995. "The Ircam Real-Time Platform and Applications", Proc. ICMC, Banff, pp. 77-83.

[2] Terrugi, D. 1991. Manuel du stage SYTER, INA-GRM.

[3] Todoroff, T. and C. Traube 1996. "Graphical NeXTSTEP Objects as FTS Clients to Control Instruments in the new FTS Client/Server Architecture", Proc. ICMC, Hong Kong, pp. 28-31.

[4] Todoroff, T. and C. Traube 1996b. "Interfaces graphiques NeXTSTEP pour la commande d'instruments virtuels d'aide à la composition et à l'interprétation de Musique Electroacoustique", Troisièmes Journées d'Informatique Musicale JIM96, Ile de Tatihou, France, pp. 98-102.

[5] Todoroff, T., C. Traube, and J.-M. Ledent 1997. "Interfaces graphiques pour la commande d'outils de transformations sonores", Quatrièmes Journées d'Informatique Musicale JIM97.