MODALYS DEMONSTRATION

Nicholas Ellis
IRCAM
1 place Igor Stravinsky
75004 Paris

Joël Bensoam
IRCAM
1 place Igor Stravinsky
75004 Paris

René Caussé
IRCAM
1 place Igor Stravinsky
75004 Paris

ABSTRACT

The IRCAM Musical Acoustics Team is not only focused on the musical instruments modelisation research but also on the computer implementation of these models and the evaluation of the musical interest that the computer implementation may have. The physical modelling sound synthesis software *Modalys* is an environment where virtual acoustical instruments may be imagined, performed and manipulated. In this paper, we present an overview of the ideas and recent developments that make *Modalys* a powerful and particular tool used by musicians.

1. INTRODUCTION

Modalys is a physical modelling synthesiser being developed for about twenty years in the instrumental acoustic team at the IRCAM. It is rooted in the research works carried out by Jean-Marie Adrien [1] [2]. The synthesis algorithm makes use of the modal formalism: any given vibration of a structure can be decomposed in a weighted sum of characteristic vibrations named modes. A mode is described by its modal shape, its eigen frequency and a loss factor, characterising the decay rate of the energy if the structure is left free to vibrate. For each sample, all the equations governing the temporal evolution of theses modes as well as the interactions (not necessarily linear) are solved in order to obtain the new state of the system. The produced sound is taken to be the velocity listened at certain points, named "accesses" in the *Modalys* terminology. A great advantage of the modal framework is the modularity it allows: each objects being consistently described by its modes, one can easily hybridise them or couple different physical domains (mechanics / acoustic e.g.). The other advantage is the natural rendering of the synthesised sounds as well as the ability to obtain typical instrumental effects: resonance in sympathy, multi-phonics, flageoletto...

1. FINITE ELEMENTS

Until recently, the *Modalys* user had either to choose his vibrating structures, be them resonating or exciting, in a limited set of objects with a simple geometry (string, rectangular or circular plate, tube...) for which the modes are analytically computable or had to explicitly give the modal data obtained by other means, by experiment for example. Recent developments [3] allow *Modalys* to create three-dimensional meshes and to compute their modes by finite elements numerical methods. *Modalys*

is able to directly communicate with the mesh visualiser *medit*, developed at the INRIA [4], allowing thus the user to visualise the mesh or to view an animation showing the structure vibrating in a certain mode. Currently, the interactions between structures are point-like but the team aims to implement spread interactions in a near future.
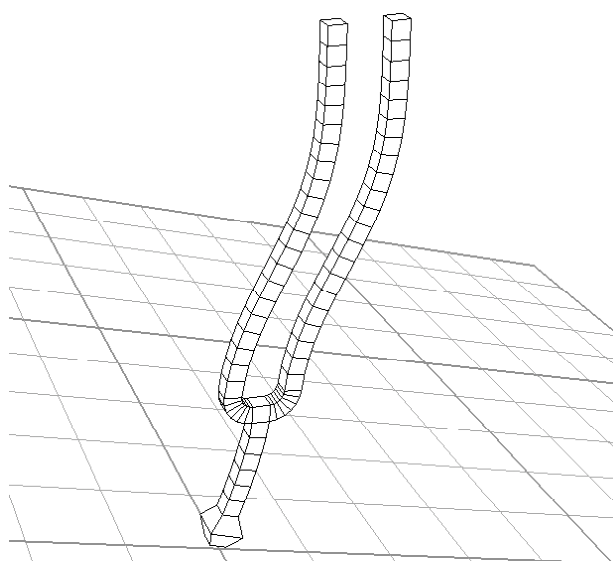

Figure 1. Visualising a modelled tuning fork in *medit*

2. INTERFACES

2.1. Kernel

The code, written originally in C, has been completely reengineered in 1991 by Joseph Morrison in an efficient and easily up-gradable kernel in C++. That language being not particularly easy to master, a simpler interface in lisp is provided. The user may define variables representing objects, accesses on these, and connections between them. It is also possible to create "controllers", which simply represent arrays of numbers evolving during synthesis. These controllers are at the heart of the user / synthesis interaction. It is through them that the user controls the force applied on the objects, the speed or the imposed position on them and are generally of the 'envelope' type. These envelopes, representing the musical gesture, are nothing less than piece-wise linear functions (break point functions, else known as BPFs) given as a list of time / values pairs.

2.2. OpenMusic

Programming in a textual language is nevertheless austere for who is not a computer engineer and that is the reason why an interface with the musical environment OpenMusic (OM) [5] [6] has been developed. Programming *Modalys* then becomes graphical, more entertaining, simpler and allows the user to benefit from the many functionality's present in other OM libraries. The basic building blocks of *Modalys* (objects, accesses, controllers, connections) are now instantiations of classes depicted by small icons which one need only to connect properly. It is possible to create user-defined function by creating sub-patches. These user-defined functions can be called during synthesis, by means of so called foreign-call controllers.
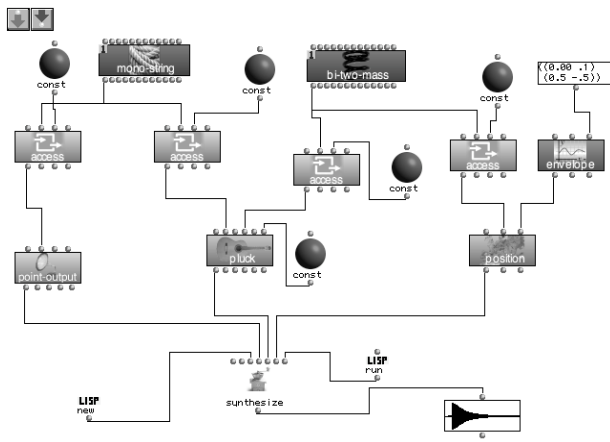

Figure 2. *Modalys* patch in OpenMusic.

2.3. OpenSound Control

The synthesis engine has been greatly optimised, which allows the synthesis to be done in real-time, in as much the load is bearable by the computer. New controllers allow a real-time interaction during synthesis: OSC, midi or audio. OSC is a communication protocol developed at the CNMAT at Berkeley [7] [8] allowing computers, synthesisers or other multi-media devices to interact one with another. OSC messages may be sent from many real-time applications, such as Max/MSP with an appropriate plug-in or PD. Real-time synthesis opens new perspectives, allows the very important feedback loop musical gesture / produced sound to take place and will certainly speed up the learning process of the musically relevant ways to "play" the virtual instruments.

2.4. Matlab

An interface with the mathematical environment MATLAB of MathWorks has been developed as well. It gives the scientific researcher a comfortable workbench to test new interaction models or objects, and allows an easier integration of these novelties in the kernel of *Modalys*.

3. FINAL NOTE

The demonstration will be illustrated with numerous examples, musical work excerpts as well as real-time demos.

4. REFERENCES

[1] Adrien, J.M. *Etude de Structures Complexes Vibrantes, Applications à la Synthèse par Modèles Physiques*, Thèse à l'Université de Paris VI, Paris, France, 1988.

[2] Adrien, J.M. "The Missing Link: Modal Synthesis" in De Poli G., Picalli A. and Roads C., *Representations of Musical Signals.*, MIT Press, Cambridge, Massachussets, USA, 1991.

[3] Bensoam J., *Formulation Intégrale Appliquée à la Synthèse Sonore par Modélisation Physique*. Thèse à l'Université du Maine, Le Mans, France, 2004.

[4] http://www.ann.jussieu.fr/~frey/logiciels/medit.html

[5] Agon C., *OpenMusic: Un Langage Visuel pour la Composition Assistée par Ordinateur*. Thèse à l'Université de Paris VI, Paris, France, 1998.

[6] Assayag G., Agon C., Fineberg J. Hanappe P, "An Object Oriented Visual Environment for Musical Composition" *Proceedings of the International Computer Music Conference*, San-Francisco, Thessaloniki, Hellas, 1997.

[7] Wright M., Freed A.,"OpenSound Control: A New Protocol for Communicating with Sound Synthesizers". *Proceedings of the International Computer Music Conference*, Thessaloniki, Hellas, 1997.

[8] Wright M., Freed A., Momeni A. "OpenSound Control: State of the Art 2003". *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, Montreal, Canada, 2003.