

# Interface Decoupled Applications for Geographically Displaced Collaboration in Music

Álvaro Barbosa, Martin Kaltenbrunner; Günter Geiger

Music Technology Group - Pompeu Fabra University  
Passeig de Circumval·lació 8 - 08003 Barcelona, España  
*email:* {abarbosa; mkalten; ggeiger}@iua.upf.es

## Abstract

In an interactive system designed to produce music, the sound synthesis engine and the user interface layer are fully integrated, but usually designed in parallel and in a modular way. Decoupling the interface layer from the synthesis engine, not only allows the use of best suited technologies and programming languages for each purpose, but also enhances the overall system flexibility. This paper discusses the idea behind a remote user interface and a processing engine that resides in a different host, taken to the most extreme situation in which a user can access the synthesizer from any place in the world using internet technology. This paradigm has promising applications in collaborative music creation systems for geographically displaced communities of user. The Public Sound Objects is an experimental system on which this concept is applied, and its currently under development at the Music Technology Group of the UPF in Barcelona.

## 1 Introduction

The idea of having an interface and a sound synthesis engine decoupled and remotely separated by a computer network has primarily been approached in the music field with the purpose of making specific software tools available to a broad spectrum of users, somehow resembling Sun Microsystems' concept of "Nomadic Computing" introduced in the early 90's (i.e. where a network user moves and his familiar work environment must follow) [1].

These tools are either dependent on special purpose hardware or based in proprietary experimental systems developed by companies or research groups. Hence these systems' topologies are normally centralized server architectures (or based in a hierarchical group of servers).

Some of the most relevant examples were developed in the last few years, such as the project started 1995, with the support from Sun Microsystems, at the *Institut de Recherche et Coordination Acoustique* (IRCAM) concerning the creation of an on-line studio [2], based on client/server Web technology. The main purpose

of this project was to provide access to some of IRCAM's sound databases and sophisticated sound-processing tools like the phase vocoder SVP.

Access to this on-line studio was primarily conceived bearing in mind in-house access at IRCAM's intranet, since high speed network communication could be provided and it was not possible for each user to have an individual work-station with the required computing power for the studio applications. Recently this project evolved to the On-Line Sound Palette application under CUIDADO's framework [3].

A similar project was started in 1997 by Ramon Loureiro and Xavier Serra [4] at the Audiovisual Institute from the Pompeu Fabra University in Barcelona, but with a slightly different scope.

The system provided a remote interface for a sound database and signal processing, but was primarily intended to be available for a broader community of users granting access to cutting edge applications, derived from research at the institute, in a simple and effective way.

With this project it was possible to have a web front end to Spectral Modeling Synthesis (SMS) [5] technique, based on Musical Sound Modeling with sinusoids plus noise, which has many scientific and artistic applications.

Yet, these systems differ from applications oriented towards music performance by the fact that they don't provide synchronous nor real-time interaction.

## 2 Virtual Music Instruments in the Network

A Virtual Music Instrument (VMI), as described by Axel Mulder in [6], aims to provide a way to control parameters of sound synthesis in an expressive and artistically meaningful way. This requires a degree of synchronicity between a user action and its effect on the sound output with a short response, ultimately converging to real-time.

In figure 1 is represented a High-Level Model of a typical Virtual Music Instrument. The communication loop introduces latency at the

acoustic feed-back. However, considering the current state-of-the-art in computer-human interaction and signal processing technology, its value can be reduced to a few milliseconds providing real-time control of the instrument.

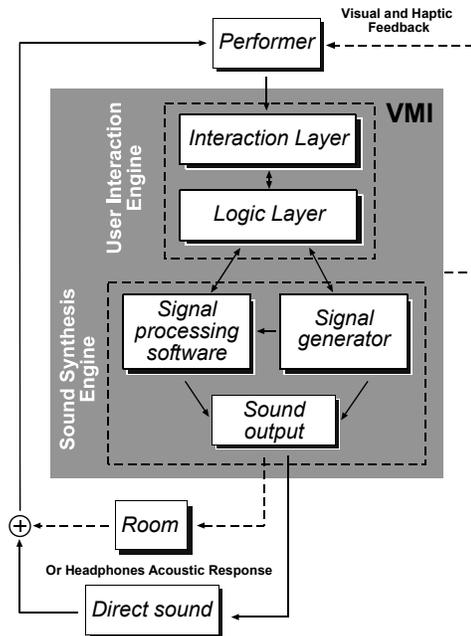


Figure 1. A model for a VMI

Nevertheless, if we separate the user interaction engine and the sound synthesis engine over a computer network, the latency will take extreme and unpredictable proportions in the feed-back loop.

The model presented in figure 2 illustrates the topology for a single user interface partitioned VMI.

It should be noticed that in this model communication from the client towards the server is based in control data transmission, however in the opposite direction the server is broadcasting a stream of digital audio towards the client.

Such architecture has advantages in multi-user setups, where different clients share the same server synthesis engine, since it will allow a unique broadcast of the music performance to all the users. This broadcast stream conveys all the contributions of the performers and it can also be accessed by a passive audience.

On the other hand, the fact that different kinds of data are being transmitted in each direction, will accentuate the unpredictable and asymmetric nature of the network delay in the system.

Since this model does not contain a client side sound synthesizer for real-time interpretation of control data, local feed-back will have considerable latency.

This introduces a big shortcoming in performance potentiality, but its a tradeoff for another important aspect, which is the possibility for the user to receive an incoming audio stream containing the results of his performance synchronized with the contribution from other users.

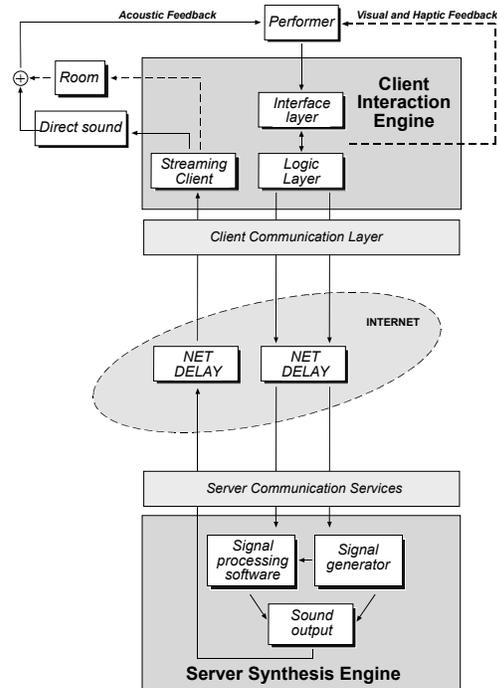


Figure 2. An Interface Partitioned VMI over a Computer Network

The Public Sound Objects Project presented in this paper, addresses the idea that, in spite of the lack of real-time interaction, it can be possible to achieve a good sense of control in an articulated and synchronous collaborative performance by exploring behavior-driven user interfaces and forms of musical expression which are better suited to the effects of network latency.

### 3 The Public Sound Objects

The Public Sound Objects project is an Internet based Collaborative Virtual Environment focused on sonic arts and music creation, currently being developed at the Music Technology Group of the Pompeu Fabra University. A preliminary specification of the system was published in [7], and the first prototype was implemented by the authors in December 2002. Conceptually, it explores the notion of a shared web space for community music creation, and of an art installation that brings together physical space and virtual

presence in the Internet. The System aims to allow synchronous interaction providing the basis for sonic joint improvisation amongst web users.

The overall system architecture was designed along the following key aspects: (a) It is based in a Centralized Server Topology supporting multiple users connected simultaneously and communicating amongst themselves through sound; (b) It is a permanent public event with special characteristics appealing both to a “real world” audience and to an on-line virtual audience; (c) On-line participants’ contribution is adequately constrained so that the overall aesthetical coherence of the piece can be guaranteed; (d) The system is scalable and modular allowing future expansion and different setups.

### 3.1 Sound Objects

In this project the raw materials provided to the users for manipulation during a performance are Sound Objects. The definition of a Sound Object as a relevant element of the music creation process goes back to the early 1960’s [8]. According to Pierre Schaeffer’s, a Sound Object is defined as “any sound phenomenon or event perceived as a coherent whole (...) regardless of its source or meaning” [9]. From a psychoacoustic and perceptual point of view, Schaefer’s definition is extremely useful, since it provides a very powerful paradigm to sculpt the symbolic value conveyed in a sonic piece.

In our system a server-side real-time sound synthesis engine provides the interface to transform various parameters of a Sound Object, which enables the user to add symbolic meaning to their performances and therefore introducing a metaphorical dimension in their expression.

### 3.2. System architecture

As shown in the illustration below, the *Public Sound Object* system is based on classic client-server architecture. The actual sound synthesis computation is handled by the server and the interaction interface is implemented on the client side. One of the main characteristics of this implementation scheme is its modularity.

The server side Sound Synthesis and Transformation Engine is designed in a rather general way allowing its versatile use for different applications. The core technology runs under Linux OS and is based on Miller Puckette’s Pure Data [10] ported to Linux by Günter Geiger, following the implementation design of Serji Jorda’s FMOL [11] synthesis engine.

The central installation will be located in a dedicated room, which can hold several people. A video projection shows a local representation of the user interface, displaying graphically the

performance of all current participants. Various loudspeakers positioned along the walls, create a spatial soundscape reproducing the sounds of the objects colliding with walls.

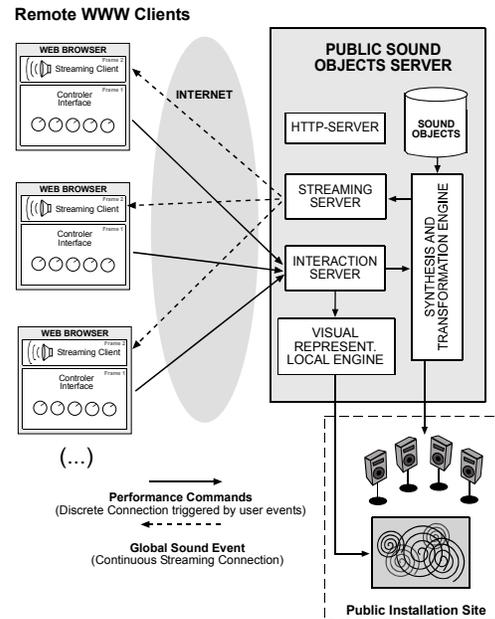


Figure 3. The PSO Architecture

On the client side the main application is a Java applet, embedded into the web interface. It provides the complete graphical user interface for the interactive control of the synthesis process, allowing the interaction with the server-side synthesis engine.

### 3.3. The User Interface

Upon loading, the user interface applet connects to the interaction server, registers and initialises a user session.

Due to the modular nature of the system the graphical user interface (GUI) can differ in different setups. Each GUI implementation, called Skin, is developed along the following requirements: (a) it should enable the user to contribute to the ongoing musical performance by transforming the characteristics of a visual Sound Object representation, sending normalized parameters to the synthesis engine over the network; (b) the interface application should be able to allow manipulation of each of the modifiers’ parameters in the synthesis engine in articulation with the specific installation site setup; (c) the GUI itself should be a behaviour-driven metaphorical interface, avoiding a flat mapping of parameters in a classical way, such as faders or knobs, and provides automatic periodical behaviour for the Graphic Objects, which can be conducted by the user.

The current implementation is the bouncing ball skin, a metaphor for a ball that infinitely bounces on the walls of an empty room. When the ball hits one of the walls a network message is sent to the central server where the corresponding Sound Object is triggered, and played through a specific source speaker and simultaneously streamed back to the user in a stereo mix of all the sounds being triggered at the moment.

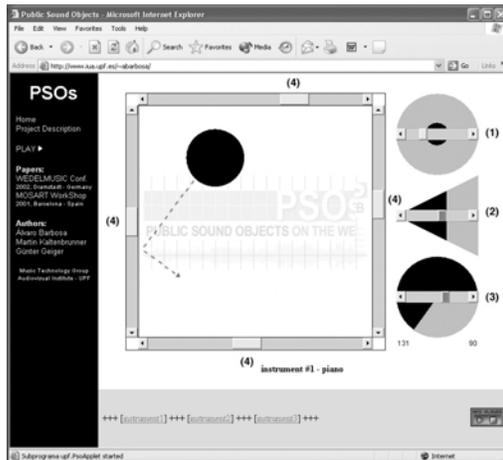


Figure 4. The Bouncing Ball Skin

The ball moves continuously and the user can manipulate its size (1), speed (2), direction (3) and each wall's acoustic texture (4). The normalized values are then sent to the server where they are mapped to synthesis parameters. The wall's acoustic texture matches the Sound Object's pitch (individual pitch values can be assigned to each wall, allowing the creation of melodic and rhythmic sound structures) and the ball size corresponds to reverberation, following a metaphor of a real room acoustic response.

## 4 Conclusions and future work

The Public Sound Objects project is still under development, however, the experiments realized so far, are quite promising.

Even with network delays going beyond 200ms, the users can achieve a good sense of high level control based on the bouncing ball behavior, having a perfect notion of what their contribution is to the overall piece and how it influences the others. On the other hand, the use of slow attack Sound Objects tends to create a *soundscape* that complies better with hard delay conditions, since it seems to fade the rhythmic synchronization requirements over time

Next step will be to incorporate the server side installation in the system and conduct more extensive experiments with increasing number of users. In future developments we will experiment with the possibility of allowing the users to upload

their own Sound Objects to the central server evaluating its musical results and different behavior driven skins.

Future work in this field at the Music Technology Group from the Pompeu Fabra University will also address alternative communication models applied to different systems, which will be based on a peer-to-peer topology, provided with client side synthesizers in order to preserve individual real-time feedback.

## 5 Acknowledgments

The Authors would like to thank Xavier Serra and Sergi Jordà for their comments and support to this project. This work was supported by the Portuguese institution "Fundação para a Ciência e Tecnologia".

## References

- [1] S. Gado and M. Clary. Nomadic Tenets - A User's Perspective. Sun Microsystems Laboratories, Inc. 1994. The SMLI Technical Report Series.
- [2] R. Wöhrmann and G. Ballet, Design and Architecture of Distributed Sound Processing Systems for Web-Based Computer Music Applications. *Computer Music Journal* 23, 73-84 (2002).
- [3] H. Vinet, P. Herrera and F. Pachet. The CUIDADO Project. 2002. Ircam - Centre Pompidou, Paris, France; e. Proceedings of ISMIR 2002 - 3rd International Conference on Music Information Retrieval.
- [4] R. Loureiro and X. Serra. A web interface for a sound database and processing system. 1997. Proceedings of the International Computer Music Conference.
- [5] X. Serra, Musical Sound Modelling with Sinusoids plus Noise. In *Musical Signal Processing*. (Ed. G. D. Poli, A. Picialli, S. T. Pope and C. Roads) Swets & Zeitlinger, 2002.
- [6] A. Mulder. Virtual Musical Instruments: Accessing the Sound Synthesis Universe as a Performer. 1994. Caxambu - Minas Gerais, Brazil. First Brazilian Symposium on Computers and Music.
- [7] A. Barbosa and M. Kaltenbrunner. Public Sound Objects: A shared musical space on the web. IEEE Computer Society Press. Proceedings of International Conference on Web Delivering of Music 2002 - Darmstadt, Germany.
- [8] P. Schaeffer, *Traité des Objets Musicaux*, 1966.
- [9] M. Chion, *Guide des Objets Sonores*. Pierre Schaeffer et la Recherche Musicale., 1983.
- [10] M. Puckette. Pure Data: another integrated computer music environment. 37-41. 1996. Tachikawa, Japan. Second Intercollege Computer Music Concerts.
- [11] S. Jordà, Faust Music On Line (FMOL): An approach to Real-time Collective Composition on the Internet. *Leonardo Music Journal* 9, (1999).