Computer supported cooperative music: overview of research work and projects at the Audio Visual Institute – U.P.F.

Sergi Jordà, Álvaro Barbosa

Email: {sergi.jorda@iua.upf.es, alvaro.barbosa@tecn.upf.es}

Music Technology Group, Audiovisual Institute, Pompeu Fabra University
Passeig de la Circumvalació 8, 08003 Barcelona, Spain

Abstract

In this paper the authors present an overview of recent work on ongoing projects at the Audiovisual Institute from the Pompeu Fabra University in Barcelona, focused on Internet collaborative virtual environments for music applications. Although presenting different strategies, they all put a special emphasis in performance and composition/production of music by groups of geographically dispersed communities of users, both in synchronous and asynchronous modes.
It is presented an overview of the concepts and developments in:

The FMOL project that approaches collaborative music composition and performance over the web in an asynchronous fashion in its original version (Jordá, 1998, 1999), and that is currently under development evolving to a new architecture with a synchronous paradigm and several relevant new features (Jordà and Wüst, 2001); The Public Sound Object project, which consists on a permanent web installation for collaborative musical performance, currently at preliminary development stage (Barbosa and Kaltenbruner, 2001).

1. Introduction

Computer-Supported Cooperative Work (CSCW) is one of the major research fields in modern society, and the current technological advances, specially in Internet computing, have allowed computer science researchers and developers to actually create collaborative tools, like textual chats, white boards, shared editors or video conference systems that are already part our reality.

Focusing on art fields, collective creation and the production of open and continuously evolving works may be two of the most appealing artistic breakthroughs the Internet can offer to creators in general and music composers in particular.

The idea of musical computer networks is by no means original; earlier implementations (although on a local area scale) date back to the late 1970s with performances by the League of Automatic Music Composers (Bischoff, Gold and Horton, 1978). Recently, several concepts and technologies from general CSCW work, have been adapted to the specificity of architectures designed for music performance and production. An example is the 1998 TransMIDI system that allows musical performers (and listeners) who wish to play together, to organize into multiple session groups over the internet using MIDI (Burk, 2000), and which has been implemented using the Transis group communication multicast communication layer for CSCW applications.

However, more than twenty years after the first experiments, collective music composition or improvisation on the net, is still at a bourgeoning state and sites and projects like *Res Rocket Surfer*, MIT's *Brain Opera*, William Duckworth's Internet based *Cathedral*, can probably still be counted with the fingers.

It is also clear that most approaches face the Internet as an element that will allow the enhancement of known cooperative music production and performance techniques.

The projects presented in this paper are not a only in the context of the recent work developed in this field, but also, they make a strong emphasis on the incorporation of the Internet paradigm as an element that will influence the musical expressiveness.

2. The FMOL system

FMOL is a system that approaches real-time collaborative music composition on the web, in an asynchronous mode. It was first developed in 1997 (Jordà, 1999) and it is currently in its third version. Using a lightweight plug-in running on top of a web browser, FMOL allows users distributed all over the Internet to work collectively on a single or on several musical pieces, sharing a common interface. It also permits new composers to modify and enlarge already existing pieces an endless number of times, while keeping at the same time the integrity of the original pieces. Its asynchronicity makes FMOL stand closer to the forum model than to the chat or videoconference paradigm. FMOL collaborative approach is based on a vertical-multitrack model (as opposed to a *horizontal-exquisite corpses* model, which would allow the pasting of sonic fragments one after the other). Its architecture allows each participant to start new compositions from scratch as well as overdubbing, modulating or processing any of the existing ones. FMOL has so far successfully been used as a virtual electronic music instrument for the collective composition of several scores for the Catalan theater group la Fura dels Baus, including the play F@ust 3.0 and fragments of the multimedia opera Don Quijote en Barcelona, premiered at the Gran Teatre del Liceu of Barcelona in October 2000.

Composers will soon benefit from advanced new features offered by the system such as user profiling, multimedia data mining and content-based retrieval. By using a collection of intelligent agents the system will have the capability to propose actions and pieces to work on, according to the users' preferences, taste and interests.

2.1 Architecture

The original system was built following a client server model. This allowed composers using the FMOL client software to log into a central server in order to download any of the pieces that were stored in a song tree-structure database. The composer was then able to work on some of the tracks of the piece with the standalone FMOL client, and send back the new version to the central server. Although the client server model has proven successful under specific circumstances, such as local, small sized productions, there exist disadvantages of using this architecture such as the installation process of the client, the redistribution and reinstallation of the client after a software upgrade or patch, and the inaccessibility of the database to curious Internet surfers.

The current version of FMOL has been built according to a three-tier architecture model, which has proven one of the most efficient architectures for Internet computing. The server side hosts a database server, responsible for all the storage and retrieval functions. In the middle tier an application server is responsible for executing all the application logic.

The application server may be physically on the same machine as the database server or on a separate one, assuming that the network connection between both machines can support a high bandwidth and low latency. Furthermore, such configurations will allow a high degree of scalability. If a large number of simultaneous users need to be supported, several application server machines can be set up, and connections to the system can be handled by a load balancing service, which will distribute the petitions across the application servers.

Universal access to the system is guaranteed by the use of a thin client. Any *wintel* personal computer equipped with a soundcard and a standard web browser will suffice for running the FMOL plug-in.

2.2 Database Tier

The FMOL system is based on a relational database. The main entity is the compositions table, which has a recursive relationship to it self, allowing for a tree like representation of the pieces, as shown in figure 1. Each piece is a node storing a scorefile that holds the data for eight real-time synthesized audio tracks, which can be played by the FMOL plug-in's audio engine. A user can pick up any existing composition, listen to it, work on it and save a new version in the database. The new version will be stored as a child node of the one the user picked.

A common problem in collaborative composition systems is intellectual property rights tracking. In our case, one of the design purposes has been to allow global access to musical collaboration. As a result many participating composers are casual Internet surfers, which makes this control even more difficult. The FMOL system implements a rights tracking option, which requires that a user is registered before allowing changes to the songs database. This control was used on the system first implementation, in 1998, after an agreement with the Spanish authors' association, SGAE, who sponsored the project and facilitated all the registration proceedings even for non-associate authors. It has not been used, however, in the 2000 implementation.

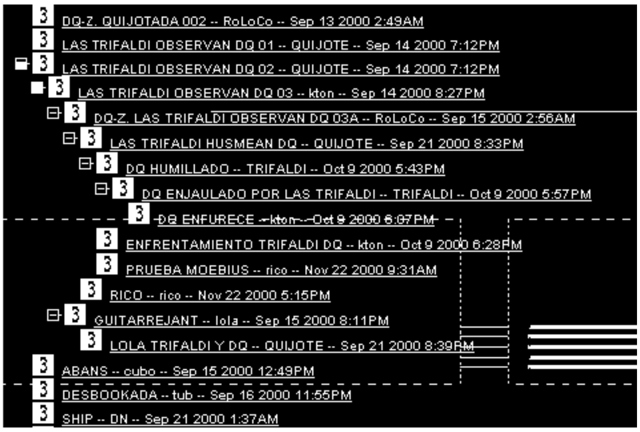

Figure 1. Screenshot showing a fragment of the compositions tree

Figure 1 shows a fragment of the compositions tree. Each line represents a node and displays the title of the piece, the author's alias and creation date. The amount of indentation reflects the depth or number of generations of the piece. In this case, several users have interacted to create up to 7 layers of collaborative work, and some of the layers (i.e. 3, 4 and 5) have different siblings.

Users are also allowed to vote on the quality of any composition. This information can help in the final selection process, and is also useful for all the advanced query features explained in section 3.

2.3 Middle Tier

The middle tier hosts the application server and the web server. These software components are responsible for running most of the program logic of the FMOL system as well as serving the presentation layer to the web browser. This includes the dynamic generation of all the web pages for user registration, profiling, voting, and most important, displaying the composition trees and managing the upload and download of compositions.

2.4 Thin client

The client tier is said to be thin because it only consists of a browser running a plug-in. The application logic is hosted mostly in the middle tier leaving the client layer only for the synthesis engine, the graphical interface and the presentation logic.

Despite the design objective of keeping the software running on the client to a minimum, there were both important esthetical and social reasons for including a specific proprietary synthesis engine, as one of the main objectives of the project was to approach experimental electronic music creation to newcomers and hobbyist musicians. In that sense, the FMOL composition and synthesis plug-in grants that everybody has access to compose, even surfers without any other audio software and no more hardware than a multimedia soundcard. This enforces an equal opportunity environment, while forcing at the same time, real-time composition and sound manipulation by means of innovative and intuitive graphical interfaces.

Although this three-tier architecture allows for different approaches which may be applied in the future without loosing any generality (as for instance the use of standard MIDI files or any other standard format which could be generated with currently available and generic software and without the need for a specific synthesis plug-in), the current synthesizer engine architecture and its graphical interfaces were in fact specially designed with this collaborative approach in mind.

The engine, written in C++ for the *wintel* platform, was meant to be a complete sound generation kernel flexible enough for real time synthesis and processing on a low-end machine (e.g. Pentium 200), that could be appealing and enriching for users with different skills and electronic music knowledge. The current version supports eight stereo real-time synthesized audio channels or tracks, each consisting of a generator (sine, square, Karplus-Strong, sample player, etc.) and three serial processors (filters, reverbs, resonators, ring-modulators, etc.) to be chosen by each composer between more than a hundred different synthesis methods or algorithms (Jordà, Aguilar, 1998).

Most important, this architecture allows any composer not only to add new sound layers to previous compositions, but also to apply further processing to any of the composition's existing tracks, modulating or distorting what other composers did, in unpredictable manners. That way, a musical idea brought by one composer can grow and evolve in many different directions unexpected by its original creator.

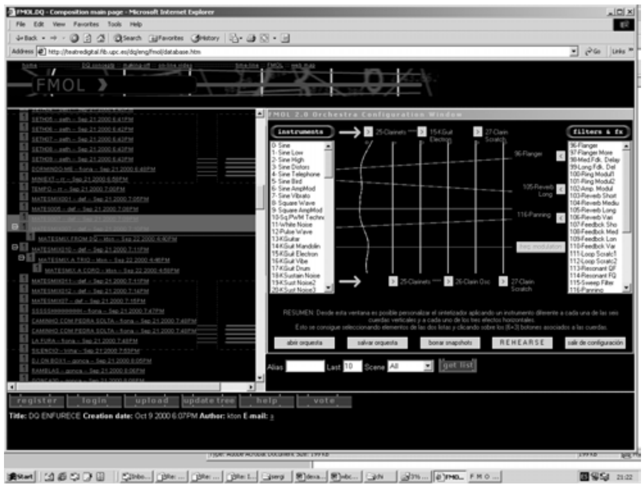

Figure 2. Screenshot showing a part of the tree database (left frame) and the FMOL plug-in configuration window (right frame)

3. Collaborative Approach New Features in FMOL

Most of the current work consists in refining the server side features of the system. The overall objective is to provide features that enforce the collective composition approach.

3.1 Real time features

Previous versions of the FMOL architecture did not implement any structures for real-time jamming. In order for several authors to collaborate on a same piece, it was required that they stored their respective contributions on the FMOL database server before any other author could continue working on them.

The problem of performing a jam session over the Internet has some constraints imposed by the current network technologies. The most relevant problem is related to the high network latencies. A note played on a computer placed on one end of an Internet connection will typically arrive with a 100-1000 ms delay to workstations on other ends of the net. This is actually unacceptable for playing most musical styles. Nevertheless, the type of music that is produced with the FMOL synthesis engine, is more timbrical than rhythmical and can therefore better tolerate timing inaccuracies, in a similar way that Gregorian chant could deal with the several seconds reverberation times produced by cathedrals. Since we consider that real-time jamming is a valid form of composing, we have decided to implement some facilities to provide this features in the newest version of FMOL, which should be considered as a complement to the current collaborative composition features of the system.

A real-time messaging server based on Phil Burk's *Transjam* protocol (Burk, 2000) will be hosted in the middle tier constituting the core of the real-time system and providing services such as the FMOL session manager, accessible through a web-based interface. Active jam sessions will be monitored, and users will be allowed to create new sessions or join any of the currently open ones, provided that the maximum number of participants per session has not been reached.

Each client participating in the jam session periodically sends to the server the events it generates. The server receives therefore the submissions from every active client. Typical data volume generated by a client is around 10 bytes per frame, where the system frame rate is configurable, but should not go below 20-25 Hz in order to keep latency and time-resolution under acceptable conditions.

To overcome the annoying latency problem we have opted to let each of the clients directly listen to what has locally been played, i.e. notes played on a client are monitored in exact real-time, without the delay of twice the latency between the client and the server. In turn, the server is periodically sending to each client a particular mix consisting of the information generated by the other participants only. Notes and events received at each client from the server are then synthesized locally. A consequence of this mechanism is that every participant will listen to a slightly different version of the collaborative piece. To minimize this effect, time stamps have been included into the generated messages, allowing for periodic server-side resynchronizations.

3.2 Pear to pear performance experiment

In October 2001, several FMOL on-line concerts took place between Barcelona and Dresden, during the *Networkshop* festival that took place in this German city. Concerts consisted of improvised duets, using a peer-to-peer alpha version of the new system. Delays attained where of the order of 100 ms using a conventional 56 kb connection, giving a very good feeling of playability and setting high expectations on the new version possibilities.

4. The public Sound Object

The Public Sound Object web installation is a project implementation that approaches the concept of co-operative composition and performance of expressive acoustic elements in real-time over the Internet. Connected users can control a server side synthesis engine through a web-based interface. The resulting sound is streamed back to each user and is also played back at the installation site. Each user's input has a direct and immediate influence on the resulting soundscape.

4.1 System architecture

As shown in the illustration below, the Public Sound Object is based on classic client-server architecture.

The main application at the server side is the synthesis engine, which will be designed in a rather general way in order to allow its further use for different applications. This means that this engine will be configured and controlled by two interfaces: a configuration interface, which initialises the general sound set-up by reading a configuration file and a control interface which allows an external application to control the various parameters of the synthesis process during runtime. The core technology of this synthesis engine will be based on Spectral Modelling Synthesis (SMS), since it provides the most flexible way to transform sounds by the modification of various parameters.

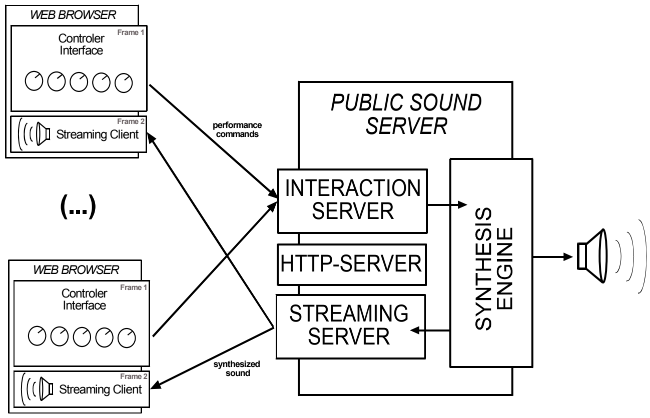

Figure 3. Public Sound Object System Architecture

The second server-side module is the interaction-server, which basically manages the various connected users, receives their input and processes those values in order to control the synthesiser engine with the results. Additional components are the streaming audio server, which broadcasts the final audio stream back to the users, and a standard HTTP server for delivering the web based interface.

On the user side, the main application will be a Java applet embedded within the web interface. This applet upon loading connects to the interaction server, registers and initialises a user session. It also provides the complete graphical user interface for the interactive control of the synthesis process. An additional frame will hold the streaming audio client for the playback of the public sound object.

4.2 User interaction concept

The user interface, which allows the interaction with the synthesis engine, will be focused on the manipulation of parameters from the sounds that are being synthesized.

The sound synthesis engine is based on Spectral Modelling Synthesis (SMS) that allows transforming pre-created sounds and it will be specially focused on the idea of morphing between synchronized periodic sound events created previously with recognizable music instruments.

Each user will be able to manipulate the ongoing musical performance by transforming the characteristics from several different tracks that are running non-stop in a synchronous way (including the possibility of muting the track).

The interaction feedback will be auditory only and it will convey the performance of all the connected users at a given moment with the respective latency due to network speed limitations.

The graphical user interface itself will be metaphorical, avoiding a direct and obvious mapping of the parameters in a classical ways (like faders or knobs). As a concept the available parameters should be able to be automated in a predefined periodical behaviour that can be changed graphically by the user.

In future development we consider the possibility of allowing the user to select and group different SMS parameters, as well as the possibility of uploading self made sounds to be morphed on the server. In this project it will be limited though in order to achieve a soundscape with an aesthetic orientation predefined by the project creators.

5. Conclusions

This paper has presented new approaches to architecting and building systems for collaborative music composition. By successfully using these design principles in a real system implementation, FMOL, we have proved the viability of our proposals. During the two periods in which this project has been on-line -January/March 1998 and September/October 2000- several hundred of composers have participated on the active creation of parts of the musical scores of two important plays by la Fura dels Baus, and a collective CD has even been released. Furthermore, we propose new ideas for collective composition environments. These are serving as a basis for current and future work.

The public Sound Object is conceptually designed in terms of its architecture and interface. At this moment the project is in an early stage of implementation and its expected to have preliminary results by the end of 2001.

References

Bischoff, J., Gold, R., and Horton, J. 1978. "Music for an Interactive Network of Computers." *Computer Music Journal, Vol. 2, No.3, pp. 24-29.*

Blackburn, s., DeRoure, D. 1998. "A Tool for Content Based Navigation of Music", *ACM Multimedia 98, Electronic Proceedings.*

Burk, P. 2000. "Jammin' on the web – a new Client/Server Architecture for Multi-User Musical Performance", *Proceedings of the International Computer Music Conference 2000.*

Jordà, S., Aguilar, T. 1998. "A graphical and net oriented approach to interactive sonic composition and real-time synthesis for low cost computer systems". *Digital Audio Effects Workshop Proceedings.*

Jordà, S. 1999. "Faust music On Line: An Approach to Real-Time Collective Composition on the Internet". *Leonardo Music Journal, Vol. 9, 5-12.*

Jordà, S., Wüst, O. 2001. "Architectural Overview of a System for Collaborative Music Composition Over the Web." *Proceedings of the 2001 International Computer Music Conference.* International Computer Music Association.

The Public Sound Object (Barbosa and Kaltenbruner, 2001) *http://www.iua.upf.es/~abarbosa/pso/pso-proposal.html*

Referenced Web Sites:

The Transis Communication System / The Caelum toolkit for CSCW: *http://www.cs.huji.ac.il/labs/transis/transis.html*

TransMIDI: *http://www.cs.huji.ac.il/~grishac/Abstracts/transmidi.html*

FMOL-DQ: *http://teatredigital.fib.upc.es/dq*

FMOL Home page: *http://www.iua.upf.es/~sergi/FMOL*

William Duckworth's Internet based Cathedral piece:*http://www.monroestreet.com/Cathedral/main.html*

Tod Machover and the M.I.T. Media Laboratory Brain Opera: *http://lethe.media.mit.edu/first-page.html*

Res Rocket Surfer site: *http://www.resrocket.com*