A System for Collaborative Music Composition over the Web

Sergi Jordà
Music Technology Group, Audiovisual Institute, Pompeu Fabra University Passeig de la Circumval·lació 8, 08003 Barcelona, Spain
sergi.jorda@iua.upf.es

Otto Wüst
Music Technology Group, Audiovisual Institute, Pompeu Fabra University Passeig de la Circumval·lació 8, 08003 Barcelona, Spain
otto.wust@iua.upf.es

Abstract

In this paper the authors propose a new architecture and new features for collaborative music composition. These design principles have been applied, without any loss of generality, to a system that has already been extensively tested on-line for the last three years, and which has allowed composers from around the world to participate in the collective creation of two important theatrical scores. They can constitute the basis for new approaches for collective composition on the Internet.

1. Introduction

Collective creation and the production of open and continuously evolving works are two of the most appealing artistic breakthroughs the Internet can offer to creators in general and to 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 [1]. However, twenty years later, collective music composition or improvisation on the net, is still at a bourgeoning state and sites and projects like *Res Rocket Surfer* [9], MIT's *Brain Opera* [8], William Duckworth's Internet based *Cathedral* [4] or the one we are presenting, *F@ust Music On-line (FMOL)* [7], can probably still be counted with the fingers.

FMOL is a system for real-time collaborative music composition on the web, started in 1997 and its third version is currently under development. 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.

2. History of a project

The FMOL project started during spring of 1997, when La Fura dels Baus, the Catalan theater group famous for its aggressive shows and performances that frequently involve audiences in an unpredictable manner, was beginning to prepare what would be its new show, *F@ust 3.0*, freely inspired in Goethe's work, and contacted us with an unusual demand: "*Given the important role symbolized by the Internet in our play, we would like part of its music to be composed by cybercomposers around the world*". Initially we did not have a clear idea of what we wanted; we just knew what we did not want: to allow people to compose tempered music on the keyboard and send us attached MIDI files via E-mail. Besides, although we felt that the project should have a fairly "popular" approach, and did not want to be therefore too demanding and restrictive about the participants' gear, we were not looking for a dull General MIDI sound, but for richer sounds and textures, that would match la Fura´s aesthetic approach and that could introduce newcomers into more experimental electronic music.

Real-time and mouse-driven software synthesis seemed therefore the natural solution for an affordable and at the same time experimental and sonically rich music creation environment. FMOL is consequently a net-based virtual synthesizer and graphical interface for real-time composition and synthesis, although, due to synchronization technical restrictions, we decided not to implement real-time communication (jamming) between

its users[1]. Its collaborative approach follows instead a half-duplex communication paradigm, which enables composers to (a) listen to already existing pieces and to (b) optionally respond and collaborate on these pieces started by other composers, as well as to (c) start new pieces at any given time.

As we will detail, its architecture 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). This allows each participant to start new compositions from scratch as well as overdubbing, modulating or processing any of the existing ones.

In should also be mentioned that FMOL experimental approach does not compete with other collaborative music projects and sites, like the popular *Rocket Network* [9], more focused toward bringing net communication facilities onto standard music production methods. FMOL emphasizes creativity and experimentation over production. We believe on one side, that the use of computers and new controller interfaces can bring a plethora of "new musics", and that, on the other side, the best way to understand and appreciate any discipline, whether artistic or not, and music is no exception, is by doing and being part of it. New and more *efficient* instruments can therefore bring new sophisticated music creation possibilities to non-trained musicians, or as Robert Rowe suggests, "let's develop *computer musicians* that do not just play back music *for* people, but become increasingly adept at making new and engaging music *with* people, at all levels of technical proficiency" [10]. The FMOL system is therefore not only constituted by a central server but it also includes proprietary synthesis engine and graphical user interfaces, providing a 100% web-based new collaborative composition environment.

3. Architecture

The first version of the 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. 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, some of the major disadvantages of using this architecture were the installation process of the client, the redistribution and reinstallation of the client after a software upgrade or patch, and the inaccessibility of the music database to curious Internet surfers.

The second version of FMOL has been built according to a three-tier architecture model, which has proven to be 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.

3.1. Database Tier

The FMOL system is based on a relational database. The main entity is the compositions table, which has a recursive relationship to itself, allowing for a tree like representation of the pieces. Each composition is represented by a node storing a pointer to 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 of the existing compositions, listen to it and work on it by overdubbing existing tracks or by adding a new track. The rework of a user is already considered to be a new composition. When the changes are posted to the database, the new composition is created as a child node of the one the user picked. This new child node will hold a pointer to the new scorefile. Therefore, the deeper a node is in the tree, the more revisions the piece will have had. It is in fact the tree structure itself which is implementing the version control. All nodes are public and the possibility that any user can modify any of the existing nodes enforces the collective composition approach.

Figure 1 shows a fragment of the compositions tree titled "3". Each line represents a node and displays the title of the composition, 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

[1] This feature will be implemented in FMOL 3.0, which will be on-line on autumn 2001.

collaborative work, and some of the layers (i.e. 3, 4 and 5) have different siblings.

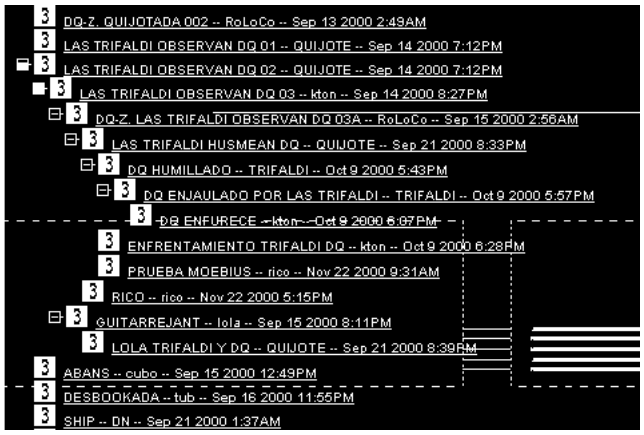

Figure 1. Screenshot showing a fragment of the compositions tree

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's 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.

Users are also allowed to vote on the quality of any composition. This information is stored in the database and can be used in various ways. First, the overall acceptance of a piece can be determined by the total number of votes scored, and this has proven helpful as an objective measure for the administrators of the system when some compositions have to be selected. Furthermore, we can take advantage from all the votes made by a single user, incorporating this information into the user's profile, which is useful for all the advanced query features explained in section 4.

3.2. 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. We have opted for dynamic generation of the web pages using Java Server Pages (JSP) and for Java Beans to store the program logic. This includes all the web pages for user registration, profiling, voting, and most important, displaying the composition trees and managing the upload and download of compositions.

3.3 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. Moreover, for each track (except for track 1) the called *generator* can in fact behave as a *parallel processo*r, which can take its input from the output of any of the lower channels (i.e. channel 5 can be configured to process channel 1, 2, 3 or 4). Each

generator or processor possesses eight control parameters, four of which can be modulated by four independent low frequency oscillators (LFO), which makes a total of 128 LFOs (4 LFOs/algorithm * 4 algorithms/track * 8 tracks) that can be active simultaneously. The type of each LFO can also be dynamically configured (sinusoidal, square, triangular, saw tooth or random)[6]. All these parameters are updated at a fixed frame rate of 48 Hz.

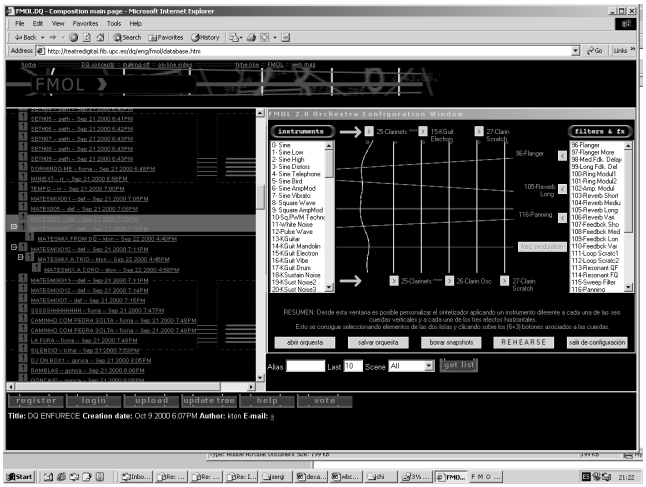

Figure 2. Screenshot showing a part of the tree database (left frame) and the FMOL plug-in configuration window (right frame)

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.

4 Collaborative approach new features

Current work consists in refining and enhancing the server side features of the system. The overall objective is to use the stored information to enforce the collective composition approach. The main techniques applied are user profiling and content-based retrieval of the compositions. This enables the FMOL system to automatically propose compositions to the users for them to work on, according to their respective preferences and inferred taste and interests.

4.1 FMOL file format and transformation into XML

Compositions in the FMOL system are stored as scorefiles consisting of time stamped commands for the real-time synthesis engine. Each scorefile has a fixed-length header and a variable part which stores the multiple tracks created by the multiple authors.

In order to do content based processing and analysis of the information, we have found adequate to have the information in XML format. This allows for easy parsing of the recorded attributes and events. It is even possible to store the XML file as a large object in the database, accessing and indexing the individual attributes. An FMOL-XML transformation component is currently under development.

4.2 User profiling

By means of user profiling [5], a system can gain knowledge about the preferences of a given user. The system can then take advantage of this information for various purposes, such as suggesting the most adequate partners for collaboration, or the most adequate musical pieces for participation in collective composition.

In FMOL, the user profile information is acquired in several ways. Through a preferences section, the user can actively enter subjective information, such as his interest in musical genre, favorite instruments, musical training and level of expertise, etc.. In addition, FMOL will monitor the user's behavior and interaction with the system. Through the compositions that a user chooses for collaboration and the votes he submits, FMOL can cluster the authors into virtual communities. Furthermore, for each of the pieces published by a user, FMOL will automatically extract and infer objective information about the composition, such as density of the notes, rhythm, melodic lines, orchestration, etc.. This profile information is stored as feature vectors that form an n-dimensional space. We are currently evaluating several of the existing techniques for performing similarity queries in such feature vector spaces.

The system is constantly being tuned towards the preferences of the users by taking knowledge of their feedback. By using its profile information, it can propose a list of pieces for the composer to work on, according to his/her preferences. After working on a piece that has been suggested by the system, the author can evaluate the quality of the proposal, and this information will be stored in the system and taken into account in its next proposal.

4.3 Content based retrieval

Another new feature of the proposed architecture is the inclusion of content-based retrieval functions. Since the musical information of the pieces has been extracted into XML structured feature vectors queries can be performed in this feature space. One of the problems to consider is that most of the extracted features are not significant to the end composer. Previous work in the area of content-based retrieval of music has used the notion of melodic contours [2] which employs the melody profile extracted from a midi file. We are therefore currently trying to exploit the properties of the synthesis algorithms to perform a mapping between low and high level descriptors. This should allow end user queries by similarity using high-level criteria, such as the notion of similar instrumentation or similar playing modes.

5. Real-time users' interaction – Net-jamming

FMOL versions 1 and 2 discarded the implementation of real-time interaction between different users, mainly because of synchronization technical restrictions, but this important feature will be finally available in the new version, thus allowing several players to share a common environment (i.e. to improvise together).

Real-time asynchronous and multipoint messaging through the Internet poses serious timing and synchronicity problems. Different implementation decisions have to be made, from a peer-to-peer versus client-server architecture to an input versus state synchronism. Latency and synchronicity needs from a musical performance point of view in general and for FMOL in particular, do not differ too much from the ones needed for multi-user action games. Typical latency values for MIDI based Internet applications can range, considering standard 56K connections, between 500-1,000 ms, which is unacceptable for most musical styles and action games. However, FMOL's compositions are more timbrical than rhythmical and can therefore better tolerate the variable delays caused by the latency, in a similar fashion as the Gregorian chant dealt with the several seconds long reverberation times of the cathedrals.

The system we are building for this purpose is a real-time messaging server based on Phil Burk's *Transjam* [3] protocol. This server resides on the middle tier and implements the FMOL session manager. A web-based interface monitors the active sessions and the participants in each of them, allowing a user to create new sessions or to enter any of the currently open ones (given that the maximum number of clients/session has not been reached).

At every frame (i.e. 48 times per second), each client sends the generated events to the server. FMOL typical data rates range from 60-180 bytes/client.second. The server, running at the same frame rate (but no necessarily synchronized) redistributes all the received messages to all the clients. We consider a priority that each user can control his/her interface with no appreciable latency. Therefore the client's own messages are treated directly before they are processed by the server, and when the server sends messages to this client, this client's messages are filtered out in order to avoid feedback.

This mechanism leads however to an unavoidable compromise, implying that every client of a session will be listening to a slightly different version of the same piece. We are currently evaluating the possibility of embedding time-stamps into the messages to allow for periodic server side resynchronizations that would minimize these differences.

We must finally point out that this real-time interaction should not be seen as a replacement, but rather as a complement of the existing collaborative possibilities discussed in the previous sections.

6. Musical and Social Implications

Two FMOL versions have so far successfully been used by hundreds of Internet composers, as a virtual electronic music instrument for the collective composition of two scores for the 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.

From January to April 1998, the FMOL first Internet-database received more than 1,100 brief pieces by around 100 composers, some of whom connected nightly and spent several hours a week creating music. One of our main goals (i.e. to conceive a musical system which could be attractive to both trained and untrained electronic musicians) was fully attained. We know now that several of the participants had no prior contact with experimental electronic music and a few were even composing for the first time, but all of them took it, however, as a rather serious game, and the final quality level of the contributions was impressive. After a difficult selection process (only 50 short pieces could be chosen and included on the show's soundtrack), and considering that a great number of interesting compositions had to be left aside, we decided some months later to produce a collective CD with a mixture of old and new compositions.

A new web with a new version of the software has been back on-line during September 2000 for La Fura´s new show, the opera DQ, premiered last October at the Gran

Teatre del Liceu in Barcelona[2]. During one month, more than 600 compositions have been submitted, and the selected ones constitute now the electroacoustic parts of an otherwise orchestral score. A third version with new features will be available in autumn 2001.

7. Conclusions

This paper has presented a new approach to architecting and building a system for collaborative music composition. By successfully using these design principles in a real system implementation, FMOL, we have proved the viability of our proposals. Furthermore, we propose new ideas for collective composition environments. These are serving as a basis for current and future work.

8. References

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

[2] S. Blackburn, D. DeRoure, "A Tool for Content Based Navigation of Music", *ACM Multimedia 98, Electronic Proceedings.*

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

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

[5] T. Fawcett, F.J. Provost, "Combining Data Mining and Machine Learning for Effective User Profiling", *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press, 1996, pp. 8-13*

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

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

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

[9] Res Rocket Surfer site: http://www.rocketnetwork.com/

[10] Rowe, R. Interactive Music Systems – Machine Listening and Composing, The MIT Press, Mas, 1992. p. 26

[2] Visit the web, download the software or learn more about the DQ-FMOL project at http://teatredigital.fib.upc.es/dq