

Architectural Overview of a System for Collaborative Music Composition Over the Web

Otto Wüst, Sergi Jordà

Music Technology Group, Audiovisual Institute, Pompeu Fabra University
Passeig de la Circumval·lació 8, 08003 Barcelona, Spain
emails: otto.wust@iaa.upf.es, sergi.jorda@iaa.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 music composers and creators in general. 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). However, twenty years later, collective music composition or improvisation on the net, is still at a burgeoning 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.

The architectures we are proposing have been implemented in FMOL, a system for real-time collaborative music composition on the web, first developed in 1997 and currently in its third version (Jordà, 1999). 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.

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 can now benefit from advanced 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 has the capability to propose actions and pieces to work on, according to the users' preferences, taste and interests.

2 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. 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.

2.1 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, 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.

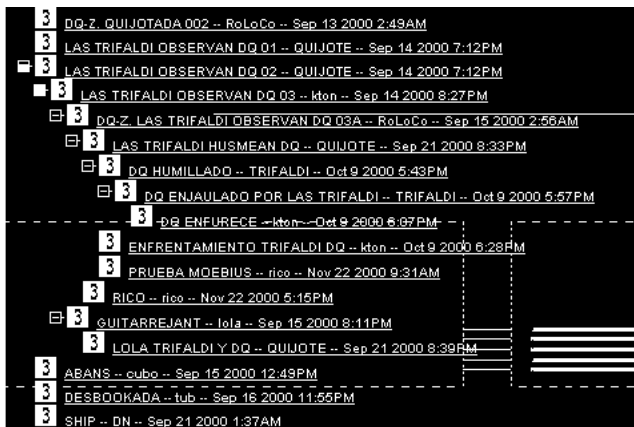


Figure 1. Screenshot showing a fragment of the compositions tree

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.

2.2 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. 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.

2.3 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 losing 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 knowledges. 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).

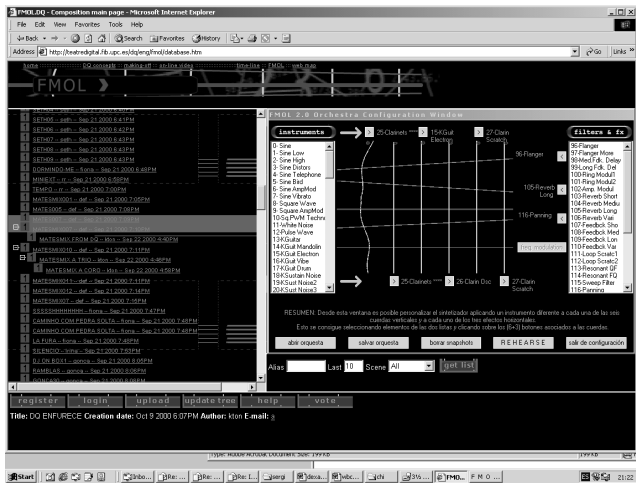


Figure 2. Screenshot showing a part of the tree database (left frame) and the FMOL plug-in configuration window (right frame)

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.

3 Collaborative Approach New Features

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. This is mainly achieved by using techniques adapted from areas such as user profiling or content-based retrieval of information.

3.1 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. In order to do any content based processing or 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.

3.2 User profiling

By means of user profiling (Fawcett, Provost, 1996), 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.

3.3 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 (Blackburn, DeRoure, 1998) 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. Nevertheless, this is still at an early stage of development.

3.4 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 500-1000ms 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 timbral 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) is hosted in the middle tier and is the core of the real-time system, providing services such as the FMOL session manager, which is accessible through a web-based interface. Active jam-sessions can be monitored, and a user can create new sessions or join any of the currently open ones, if the maximum number of participants per session has not been reached.

Each client participating in the jam-session periodically sends the played notes and events to the server. We have set a frame rate of 48 times per second considering the typical data volume generated by the client which is between 60 and 180 bytes per second. The server receives the submissions from every active client.

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. The notes and events received at the client from the server are resynthesized locally. A consequence of this mechanism is that every participant will listen to a slightly different mix of the collaborative piece. To

minimize this effect, time stamps have been included into the generated messages, allowing for periodic server-side resynchronizations.

4 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. 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.

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*.
- Fawcett, T., Provost, F.J. 1996. "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, pp. 8-13.
- 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.

Referenced Webs:

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