

A web interface for a sound database and processing system

Ramon Loureiro

Xavier Serra

Audiovisual Institute, Pompeu Fabra University

<http://www.iaa.upf.es>

{rloureiro, xserra}@iaa.upf.es

[published in the Proceedings of the International Computer Music Conference 1997]

Abstract

The World Wide Web has brought the possibility of distributing multimedia objects through the Internet, becoming a very interesting platform for the development of innovative music and audio applications. Apart from the explosion of commercial plug-ins and compression techniques aimed at real-time transfer of music and audio data, there are some projects that go beyond that by attempting to use Internet as a musical production environment. In this article we will discuss our view on the technological requirements for an ideal Web-based studio and as a particular application we will show a Web front-end to the Spectral Modeling Synthesis (SMS) system.

1. Introduction

A musical production environment based on the Web, a Web-based studio, is built on the idea of a network application that takes advantage of the client-server architecture of the Web. A host machine is used as a sound database and a processing server and the control is in the client machines. A user can access and use the processing and storage potential of a remote studio in a friendly and interactive manner.

The traditional brick wall separation between client and server can be broken with the use of Java [3] or ActiveX [4], transferring part of the processing to the client machines in the cases where it might be more efficient to perform some of it locally. A further step towards flexibility is to have a choice as to where to perform a particular process, which should be transparent to the user and done depending on the load of the Net and the configuration of each particular local machine. Thus having a dynamic configuration of the client-server architecture. We can also improve performance by organizing the server data as a distributed database, where the data is stored in several hosts and the main server deals with its unified organization.

To the best of our knowledge there is no finished system with these characteristics. However, several projects have been started in this general research area [5][6][7]. We will demonstrate a project being developed at the Audiovisual Institute of the Pompeu Fabra University that explores different Web based

capabilities for the remote usage of a sound processing system. In particular we will present a distributed database and several Web interfaces [1] to a spectral based system for the analysis, transformation and resynthesis of sounds [2]. The database includes a collection of instrumental and vocal sounds plus sound effects that we have recorded and organized into the server. From this database the user can select and process any sound, and transfer, out of real time, the results to a client machine. Currently available sound transformations include: sound morphing, time stretching, pitch changes and various spectral transformations. The interaction with the user is based on menus and forms that automatically generate new interfaces depending on the user choices. The user can also submit sounds to be processed by the system.

In this article we will briefly discuss our view on the technological requirements for an ideal Web-based Studio and introduce the system that will be demonstrated at the conference.

2. A Flexible Network Architecture

Traditional computer networks include a main frame acting as a host and serving files and processes to dumb terminals. Nowadays, we find that the typical clients accessing a network are powerful personal computers and that most of the machines acting as servers are workstations with a compute power

similar to the clients. Thus, these clients could take a more active part in a particular application.

In many Web-based applications the processing power is migrating towards the clients, even though the client-server software is still the same than several years ago. Java has permitted an advancement towards client-side computation, allowing the program to be transferred to the client and executed there. The problem comes when the program is large and the connection to the network is not fast enough for an efficient transfer. ActiveX is, in that sense, a better solution, because the program travels only once to the client and is installed there for future work sessions. Nonetheless, ActiveX is still a solution that restricts the possible client platforms.

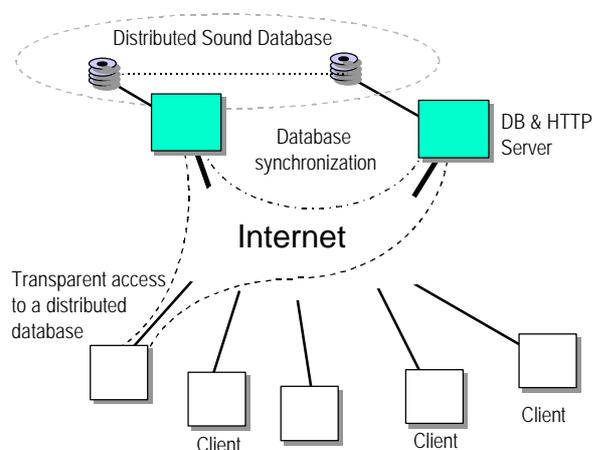


Figure 1: Client-Server architecture for a Web-based studio.

The “cgi-bin” protocol, part of the *http* server and used by many Web servers, offers simple but powerful tools that permit to track the state of the network. With this information the server can decide at each moment, dynamically, whether itself or the client will function as a processing server. The system will then commute between HTML forms and Java applets.

Broadband networks, with guaranteed bandwidth and constant end-to-end delay, are being built gradually in most countries, incorporating emerging technologies which profit from optical fiber cabling. This type of connections give access to applications such as real-time audio and video and other real-time applications which require a high degree of continuity.

Therefore, current software and hardware developments are taking us towards very powerful and flexible network architectures with a lot of potential for innovative music and audio applications.

3. SMS

The current version of the SMS software includes a set of spectral-based techniques for the analysis, transformation and synthesis of sounds [2][8]. It is written as a set of C++ classes with which we can obtain several types of spectral representations from a sound. These representations can be transformed, and then we can synthesize high quality sound from them. Even though the current system includes a graphical interface for Windows 95, it is completely independent from it. All the processing C++ classes are portable to other platforms and other interfaces can be easily built for them.

SMS offers a powerful signal processing basis for dealing with sounds in a Web-based studio situation. It offers: (1) high level representations of sound without any loss in sound quality, (2) a unified representation, control, and synthesis engine for several spectral sound models, (3) many sound transformation possibilities, (4) efficient high quality audio coding, and (5) useful tools for sound processing and music synthesis applications.

4. Sound Databases

A large and well organized sound database is a very important component in a music production studio. Most multimedia databases permit to assign keywords and/or descriptions with each sound, and the search is done by looking this textual information. This is a major limitation due to the natural complexity of sound and music, and good alternatives are based on content-based retrieval systems [9]. With these systems we can search a database by specifying acoustic or perceptual attributes which are obtained directly from sound analysis.

Most of the useful attributes of a sound that we would like to search for in a database can be obtained in the spectral domain. By having, in a database, the sounds stored in the format that results from the SMS analysis we could build a powerful, flexible and efficient content-based retrieval system.

5. Distributed Databases

A flexible client-server architecture is a good solution to the problem of serving processes on the Internet depending on the network load. This load also affects the access to the database of the server, that is, the

transfer of sounds. A simple solution to this, even though is a bit hard to manage, is the idea of a distributed database (Figure 1). A database distributed over several hosts, each one located in a different country is a viable solution. The database servers would synchronize their contents during times of low network load and the clients would be able to choose their “closest” server. Therefore the information would be duplicated in different geographical locations.

6. A Web-based Studio

We are aiming for a production environment that is powerful and flexible. Powerful in the sense of having access to a large sound database and a large compute engine that could run sophisticated and varied signal processing algorithms. Flexible in the sense of being able to update its configuration as new developments are incorporated, and add new sounds, algorithms, or interfaces.

The traditional music studio is based on many hardware devices, each one specific for a particular task. It might be quite powerful for particular processing needs but not very flexible. In turn, software based production environments are more flexible, especially if they can be extended by adding new signal processing plug-ins, but most people do not have large enough computers to run complex algorithms on them.

The Web-based studio offers interesting alternatives to the traditional studio. Such a studio would consist of (1) a large main frame functioning as a processing server and main http server, (2) a large sound database with a content-based retrieval system which could be duplicated into several servers, (3) a large collection of signal processing algorithms for all kinds of sound transformations, and (4) downloadable client-side graphical interfaces and signal processing applets.

The network connection to the clients should be the fastest possible, specially if we want to offer real-time interaction. However it should be based on a flexible client-server architecture that could adapt to each type of connection and client platform.

A client would connect to the Web-based studio and use all the processing and storage potential of the remote site in a friendly and interactive manner. Depending on the computer used as client we could transfer more or less of the processing software to the local machine.

7. SMS FrontEnd

Our sound database and processing system based on the Web, SMS FrontEnd, evolves towards the ideas mentioned and offers an interesting experimentation platform for most of the SMS techniques.

With the SMS FrontEnd the user accesses a database of natural sounds, instruments, and voices, from simple HTML forms generated at run-time. We can either start by analyzing a sound to obtain the corresponding spectral representation, or start directly from the database of analyzed sound. We can also upload our own sounds to be analyzed.

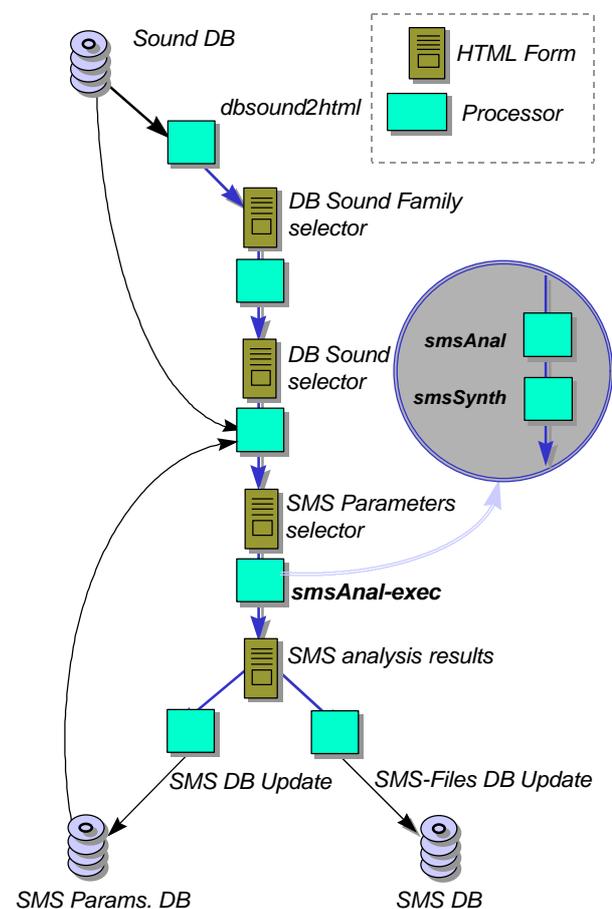


Figure 2: Diagram of the analysis part of the SMS FrontEnd.

If we start from the database of sounds, we first select a sound from the organized library. Once chosen, the system displays its time domain waveform and the analysis parameters that are best fitted for that sound. The default parameters for each sound are entered in a training phase by an expert user and stored in a database that can be updated by any authorized user. A regular user can change any of the default parameters and send the analysis request to the

server. The analysis is done (transparent to the user), in one of the four SGI machines of our local network. The system chooses the one with the minimum load. The analysis results are shown, together with a resynthesized version of the original sound. These results can be downloaded to a client machine.

The analysis output, an SMS file, can become part of the database of analyzed sounds, and we can transform and synthesize a new sound from it. A wide range of transformations are available including: sound morphing, time stretching, pitch changes, and various spectral transformations.

As an intermediate solution towards a client-side computation, the SMS FrontEnd makes simple checks, using Java applets, of syntax and value ranges of the parameters specified by the user in the client machine.

The most compute intensive part of the system is the sound analysis. The synthesis is quite efficient and it could easily run on the client machine. Thus we would transfer the spectral representation of the sound to be synthesized in the local machine. Java is still not quite powerful and efficient enough for implementing the synthesis process and a current solution is to use the C++ synthesis engine of the SMS system as an ActiveX applet. This applet is transferred the first time a synthesis is requested, and it will be transferred again only when a change is made to the synthesis engine. Hopefully Java will become more efficient in the near future.

8. Conclusion

We have presented some ideas for building a Web-based musical production system and shown an example of a work in progress. There is still a long way to go before we can actually have fully functional and musically useful systems, but the direction is quite clear and there are many people from different centers working towards that. We will do our best to achieve this goal.

9. Acknowledgements

We would like to acknowledge the contribution to this research of other members of our group: Jordi Bonada, Perfecto Herrera, Josep Maria Solà, Jaume Soler, Esther Guerra and Xavier Amatriain.

References

- [1] Audiovisual Institute. 1997. SMS Web site. URL:<http://www.iaa.upf.es/~sms>.
- [2] Serra, X. 1996. "Musical Sound Modeling with Sinusoids plus Noise", in G. D. Poli, A. Piccialli, S. T. Pope, and C. Roads, editors, *Musical Signal Processing*. Swets & Zeitlinger Publishers.
- [3] Sun Microsystems Inc. 1996. Java. URL:<http://www.sun.com/java/sw.html>
- [4] Microsoft Corporation. 1997. ActiveX. URL: <http://www.microsoft.com/kb/articles/q154/5/44.htm>
- [5] IRCAM. 1997. Studio en ligne. URL: <http://www.ircam.fr/produits-real/multimedia/studioligne-e.html>.
- [6] Glasgow University and others. 1997. NetMUSE. URL:<http://www.music.gla.ac.uk/HTMLFolder/Research/NetMUSE.html>
- [7] Casey, M. and P. Smaragdis. 1996. "NetSound", *Proceedings of the ICMC 96*. (NetSound homepage, URL:<http://sound.media.mit.edu/~mkc/netsound.html>)
- [8] Serra, X., J. Bonada, P. Herrera and R. Loureiro. 1997. "Integrating Complementary Spectral Models in the Design of a Musical Synthesizer", *Proceedings of the ICMC 97*.
- [9] Blum, T., D. Keislar, J. Wheaton, and E. Wold. 1995. "Audio Databases with Content-Based Retrieval", *Proceedings of the ICMC 95*. (Demonstration in URL: <http://www.musclefish.com/cbr.html>)