Solutions for Distributed Musical Instruments on the Web

Jônatãs Manzolli ¹,³, Márcio O. Costa ¹,², Fernando L. Ramos ¹,², José E. Fornari ¹,⁴, Dan Sharoni ¹

¹Interdisciplinary Nucleus for Sound Studies - NICS
²Institute of Computation - IC
³Faculty of Arts, Music Department - DM/IA
⁴Faculty of Electric Engineering
University of Campinas – UNICAMP, BRAZIL

[jonatas | marcio | Fernando | fornari | dsharoni]@nics.unicamp.br

Abstract

This paper presents new applications for the creation of music using the JAVA platform on the Internet. We describe the implementation of two new musical instruments named Rabisco and Cordas Virtuais. The new instruments can be used in interactive performances where MIDI Server receives several streams of MIDI data from several clients and transmits mixed sound back to the clients, thus allowing human and machine to cooperate in a new use for the Web: a place for distributed musical performance.

Key words – Interactive Music, Computer Music, MIDI, JAVA Sound

1 Introduction


In parallel, “Interactive Music System” (IMS) has been studied: Rowe (1993) defined interactive music systems as those whose behavior changes in response to music input. In this paper we present a research that uses the Internet to expand the motion of IMS. We studied how JAVA class implementation can be used in order to create new kinds of sonic interactions. We built “Distributed Music Instruments” (DMI) to allow musicians to play together and improvise on the Web. In line with Burk (2000), we developed a Client/Server architecture where the Java based MIDI Server imports several MIDI events over the Internet from clients and exports the mixed sound back to all clients. This tested
interactive architecture allows human and machine to cooperate in a single
distributed musical performance.

The first section presents the concept of DMI and the associated ToolBox
allows real time control of instruments such as the DMI named Rabisco used to
create stream of MIDI data driven by drawings on the computer screen. The
following section presents Cordas Virtuais, another DMI in which Java classes
were used to create virtual fretted-string instruments. The final section presents
results and applications in which these two DMIs were used.

2 From a Concept to a ToolBox

2.1 Statement of Problem and Purpose of Study

The World Wide Web (WWW) is not only a place to store music recordings. It
is a unique, multi-dimensional, complex and undiscovered medium, worth
exploring to find the new paradigm of creating and constructing new music. In
this research we propose that one way of accomplishing our objective is to build
a "Virtual Studio" - an environment suited for creating musical compositions,
Interactively, on the web.

2.2 Description of Terms

The concept of “Distributed Musical Instruments” (DMI) was used to focus our
efforts on exploring the Internet as a new musical media.

The vision is that of a large chat-room for distributed musical instruments on the
web. This “musical chat-room” is a way of gathering people together in cyber-
space and provide them with a medium where they can interact with each other
using DMIs. This research is dedicated to make the users virtual musicians
performing “jam-sessions”.

- Using pre-defined Java class abstractions for instrument behavior, users
can design their own virtual musical instrument.

- The user can be a musician performer, to play that virtual instrument,
and using all sorts of human skills (musical, performance and anatomic).

In order to achieve this goal, the first step was to construct a multi-tasking
Toolbox to control multiple streams of MIDI data in real time (Costa &
Manzolli, 2001). A Note Collector was developed to manage independent
sequences of MIDI data, and keep track of MIDI events, thus controlling
temporal sequences of MIDI data in real time. Every MIDI event is controlled
by two ‘Note On’ commands: one that points to the note starting time and the
other to the note dump. Further more, high level musical structures were
developed such as chords and arpeggios and all these methods were packaged in
a class we named Tvoz.
3 - *Rabisco* - Drawing Sounds on the Web

*Rabisco* provides an graphical Pad look-alike, where one can “*draw a sound*” that he/she wants to listen to. Up to four different voices are provided to the user. Each voice is driven by a set of musical parameters such as *tempo*, *rhythmic pattern*, *volume*, *musical scale*, *start on/off* and *instrument type*.

3.1 The Technology and The Design

*Rabisco* is based on the MIDI protocol and is implemented using the Java2 sound package. The motivation to use Java is its inherent platform independent characteristic. It was used on various operating systems with good integration on the Internet. *Rabisco* produces MIDI events, delivers them to the Java MIDI synthesizer that in turn generate sound without the need for standard MIDI on a sound card. Using the general MIDI Standard for instrument assignment, *Rabisco* can create and perform virtual pieces. *Rabisco* uses a Client/Server architecture where each musician plays one Rabisco drawing board at a client site (Figure 1 shows the Network configuration planned for one interactive performance).

![Network configuration diagram](image)

Figure 1 – Client/Server architecture used in a distributed musical performance.

3.2 Interaction Over The Network

Two versions of the program have been developed: one is a limited functionality applet version and the other is a full server side application with more resources for music composition.

The applet version is smaller for fast web page down/load. It has a limited feature set, due to applets security limitations and browser Java 1.4.
compatibility issues. Anyone with a browser can play the Rabisco applet on (www.nics.unicamp.br/~marcio/rabisco).

The server application version is full-featured and allows a Java MIDI Server to import several MIDI streams over a network and stream out to all clients the mixed audio. That way, various Rabisco clients, in different remote locations, can cooperate in a virtual musical performance.

3.3 Scales & Graphical User Interface

The Graphical User Interface was implemented using Java swing package that provides frames, events listeners and other advanced features not available in the AWT package. The 2 dimensional space on the drawing pad is linearly mapped to the range 0 to 127 integer, discrete values, that corresponds to the MIDI Table for Note and Velocity. X-axis was mapped to note and Y-axis to velocity. The Note Map was originally developed to assign notes in the chromatic scale, but later other scales were implemented as well. A map filter was created that separates note events in pitch classes and octaves. The first one was related to predefined scales patterns and the second were mapped according to the MIDI Table. The following scales were implemented: Chromatic, Major, Minor, Dorian, Pentatonic, Hexatonic, Blues, Aeolian, Mixolidian

![Figure 2 - Rabisco Graphic Interface](image)

4 - Cordas Virtuais - Fretted Instruments on the Web

Using the basic concept of Object Oriented Methodologies – Class Inheritance – Fretted String instrument abstraction was created. In this environment, for instance, abstractions for right hand gestures were created. A Fretted instrument was built by assembling a set of parameterized instances of the abstractions, that inherit the gestures and adds instrument specific characteristics. Using Cordas Virtuais, it is possible to imagine someone creating a 20-string instrument, each one associated with a different MIDI program and performed by a 7-finger hand. see www.nics.unicamp.br/cordasvirtuais.
4.1 Class Development

Java2 provides interfaces and classes for I/O, sequencing, and synthesis of MIDI data. We used this API to implement fretted instrument characteristics like number of strings and frets, string tunings and the right hand rhythmic actions. To model the musical gestures, we adopted the point of view of a right-handed instrumentalist. The left hand controls the fret choice while the right hand performs the “attack” on strings. We used the tablature notation to describe the left-hand actions. It allowed a spatial index of a music sequence providing a precise indication of the fret and the string to be played.

The right hand movements that control the rhythm were described by what we called Play Style. This is a class abstraction related to the instrument temporal control. Micro-rhythmic structures such as those found on the Spanish Rasgueado Style were implemented with success, (see www.nics.unicamp.br/~fernando/rasg/). The main classes were divided in Instrument and Performer packages. The Instrument package contains the Synthesizer, instrument name (MIDI program), number of strings (individual or grouped strings). The Performer package contains representations of the table and the Play Style.

4.2 Strings, Frets and Graphic Interface

Starting the research we developed a testing graphic interface that can be accessed at www.nics.unicamp.br/~fernando/cordas/. Later we studied the Play Style interface for micro-rhythmic manipulation. It consists of an interaction area for each string, where the horizontal axis determines the perceptual duration (the moment where the finger touches the string), and the vertical axis represents the attack intensity (finger’s velocity), www.nics.unicamp.br/~fernando/cordas/interface/

5 Conclusions: Results, Applications and Future Work

The Rabisco applet application is currently used in the PGL project (Partnership in Global Learning). Rabisco proved to be a fun way of causing children to enhance their sound perception and musical skills, see www.nics.unicamp.br/rabisco/pagina_rabisco.html. The Rabisco application version has been used in the Roboser and Ada projects. In this case, Rabisco creates sound patterns to be controlled by a large neural network system and these musical Style Files composed with Rabisco will be presented at the Swiss EXPO 2002, see (Wassermann et al, 2000 and www.ini.unizh.ch/~expo/2_2_3_0.html).

Cordas Virtuais applets were used among Web users. Musicians improvised, using individual virtual string instrument. We found an expressive usage for this new class of musical instrument in mixed musical performances using virtual Cordas Virtuais as well as acoustic instruments with high degree of satisfaction.
The next steps will be to enhance Rabisco graphic tools for controlling tempo changes, read and write MIDI Files and to provides a local echo for remote performance applications using the MIDI Server.

Cordas Virtuais will be applied to provide exchange of fretted instrument features and performance styles through the web. A database of tablature and Play Style will be constructed in near future.

Further research is also taking place at NICS, applying the methodology of Evolutionary Sound Synthesis (ESSynth)(Manzolli et al, 2001) algorithms to standard Java Wave-Table. It is well known that wave-table engines work with digital sound patterns sampled from actual musical instruments. ESSynth may be used to literally bring these sound patterns back to live, by applying genetic operations in three of their psychoacoustics characteristics: loudness, pitch and spectral distribution, based on user needs, and applied by the MIDI protocol as a parametric control.

6 Bibliography


7 Acknowledgments

The authors wishes to express sincere appreciation to the Brazilian Research Agencies CNPq and FAEP/Unicamp for their financial support to our research.