Virtual Studio: Distributed Musical Instruments on the Web

Fernando Lindner Ramos[1], Marcio de Oliveira Costa[1], Jônatas Manzolli[2]

[1] Interdisciplinary Nucleus for Sound Studies – NICS
Institute of Computing, UNICAMP, BRAZIL

[2] Interdisciplinary Nucleus for Sound Studies – NICS
Music Department, Arts Faculty, UNICAMP, BRAZIL

{fernando,marcio,jonatas}@nics.unicamp.br

Abstract: *Our objective was to build a "Virtual Studio", an environment suited for creating musical compositions, interactively, on the web. Starting from this point, we studied the implementation of new computer based music instruments for distributed performance on the Web, called here as DMIs. We took advantage of the recent Java2 implementation to create a general model for developing interactive musical performance among Internet users. The DMIs were used in interactive performances where a MIDI Server receives several streams of MIDI data from several clients. In this paper, we present and analyse the performance of two DMIs: the first one is called "Rabisco" which allows the user to draw sound trajectories on the Web; the second, called "Cordas", is a fretted-string instrument implemented using class abstractions. In both examples, there is a heavy usage of Java Objected Oriented packages and they run in any browser supporting the current Java Virtual Machine (JVM) across the Web. We discuss the concept of each instrument, present the system implementation, results and discuss the research next step.*

Keyboards: Internet, Distributed Musical Instruments, Interactive Composition

1. Introduction

The World Wide Web (WWW) is not only a space to store music recordings. It is a unique, multi-dimensional, complex and undiscovered medium, worth exploring to find the new paradigm to creating and constructing new music. Besides, there is a crescent interest in using the Internet as media for expression in Arts. Nevertheless, the use of Internet as a source for musical database is already established and explored, although the sound domain is not fully explored yet in its creative potential. An example is that there are lots of sites with thousands of MP3 recordings. Further, this intensive music piracy led to questioning the future of the music industry and copyright polices.

On the other hand, the Internet can be used in a creative way. Instead of making database of recorded music, it can be a way of producing new music interactively. There are many technical problems still to be solved in order to have a real interactive musical performance across the Web. Mostly, the latency of the Internet produces synchronization problems among running applications. The research presented here, studies this theme in the direction of exploring the Internet as composition media and uses the man-machine interaction as a way of producing new sonic complexity, amplifying the use of Internet in the direction of musical creativity.

The exploration of a network as a musical medium can be related to the work of a musical ensemble named "The Hub". In this case, a set of musicians played together using a local network was able to exchange MIDI data in real time. Most of the efforts of this pioneer group were in the direction of create new musical protocols for distributed musical performance [Bischoff et al 1985].

Recently, much research has been done to explore the musical potential of the Internet as a new music media. Jordá (2000) presented an approach to real-time collective composition on the Internet. As mentioned above, Jordá also commented that concepts such as authorship and copyright will necessarily have to evolve and adapt to a new reality when the Internet is used as a new musical medium. Others implemented local area network music installation [Brown 2000], where the software uses Afro-Cuban musical concepts as a model of creating an interactive drum machine. Burk (2000) developed Client/Server architecture for multi-user musical performance, while Helmuth (2000) discussed several host configurations to allow music performance on the Internet. Complementary, Hwang and Kim (2000) discussed the concept of "*Virtual Musical Environment*" (VME). There are examples over the Internet where they are used to music applications of the Java Virtual Machine (see www.transjam.com/ or http://music.calarts.edu/~tre/JavaMusic.html the Java Musical Projects).

In our research we explored the notion of musical interaction using a concept named as *Distributed Musical Instruments* (DMIs). In our systems, class abstractions are related to musical gesture and the mouse and graphical interfaces are used as input. DMIs were developed as Java application or applets. We created and studied *Rabisco* and *Cordas*: the first one is presented on two versions and the second is essentially an applet instrument. We studied a graphical paradigm for *Rabisco* and instrument paradigm for *Cordas*. The DMIs implementations were based on a Toolbox created to enable multi-task MIDI stream control [Costa and Manzolli 2001]. Independent sequences of MIDI data were generated and managed by a *Note Collector* (NC) simultaneously. The NC 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.

In the next sections some starting points about the projects will be presented, including some explanations about the development of DMIs applications and the use of Java as the chosen implementation language. Subsequently, each program implementation details, results and conclusions will be shown individually.

2. MIDI Server Implementation

We developed a Client/Server architecture where a Remote Method Invocation based MIDI server received several MIDI events over the Internet [Manzolli et al 2002A]. RMI provides the mechanism by which the server and the client communicate and pass information back and forth. The DMIs generated MIDI events, delivered them to the Java MIDI synthesizer that in turn generated sound without the need for standard MIDI on a sound card. Using the General MIDI Standard for instrument assignment, DMIs produced MIDI events in real time. It was possible to perform interactive pieces where each musician played one DMI using the mouse at a client site. The server allows importing several MIDI streams over the Internet. That way, various *Rabisco* clients, in different remote locations, can cooperate in a virtual musical performance. A representation of the MIDI server environment can be seen on **Figure 1.**

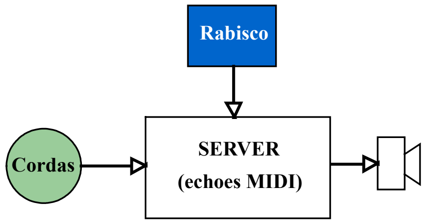

Figure 1. MIDI Server

For the moment, the server is just a remote object, but using redefinition we can change the way notes can be played, like creating a buffer and normalizing the delay and tempo for all notes. **Figure 2** presents a class diagram of the server.

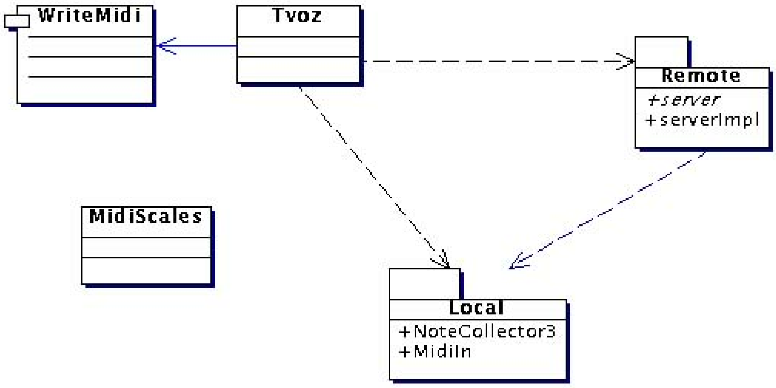

Figure 2. MIDI Server Class Diagram

3. *Rabisco* - Drawing Sounds on the Web

Rabisco provides a 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, starts on/off and instrument type*. **Figure 3** shows the *Rabisco* graphics interface in an interactive performance [Costa et al 2002].

3.1 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 two-dimensional space on the drawing pad is linearly mapped to the integer range 0 to 127, discrete values that correspond 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 that separates note events in *pitch classes* and *octaves* was created. The first one was related to predefined scales patterns and the second was mapped according to the MIDI Table. The following scales were implemented: *Chromatic, Major, Minor, Dorian, Pentatonic, Hexatonic, Blues, Aeolian and Mixolidan.*

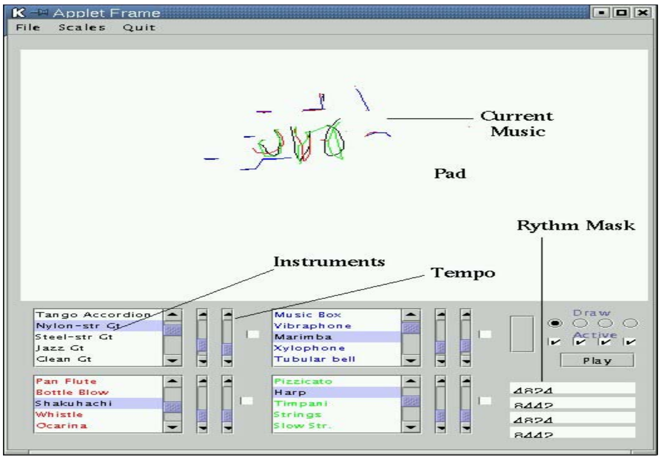

Figure 3. *Rabisco* Graphic Interface

4. *Cordas* – Fretted String Instrument

Using the basic concept of Object Oriented Methodologies (Class Inheritance), a Fretted String instrument was designed and abstractions for right hand gestures were implemented. This fretted instrument was built by assembling a set of parameterized instances that inherit the gestures and add instrument specific characteristics.

4.1 Class Development

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. **Figure 4** presents a simplified class diagram of the system. 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 *Tablature* and the *Play Style.*

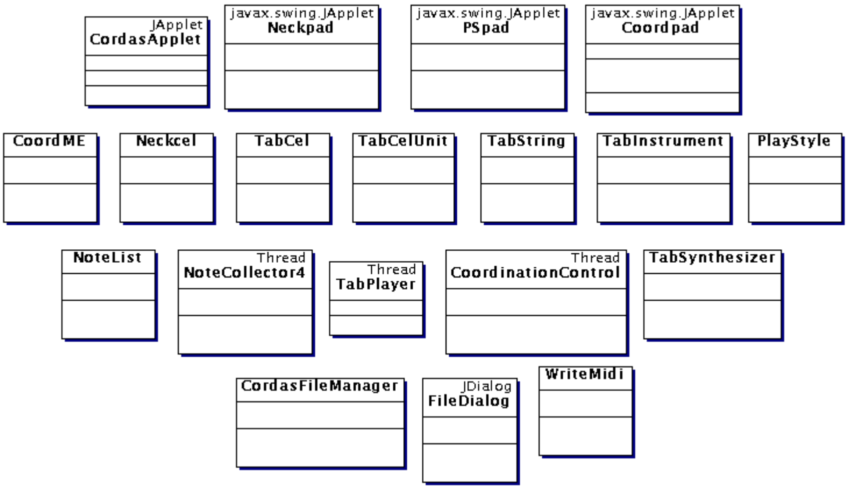

Figure 4 – *Cordas* simplified class diagram

4.2. *Cordas* Implementation

The right hand movements that control the rhythm were described by the *Play Style* interface and micro-rhythmic structures such as those found on the *Spanish Rasgueado* were implemented with success (see http://www.nics.unicamp.br/cordas/). It consists on an interaction area for each string, where horizontal axis determines the perceptual duration (the moment where the finger touches the string), and the vertical axis represents the attack intensity (finger's individual velocity).

4.2.1 Graphic to Play Style Interface

Starting the research we developed a testing graphic interface that can be accessed at (http://www.nics.unicamp.br/cordas/v1/). Later we studied the *Play Style* interface for micro-rhythmic manipulation, and the base for the actual interface was developed (see **Figure 5**).

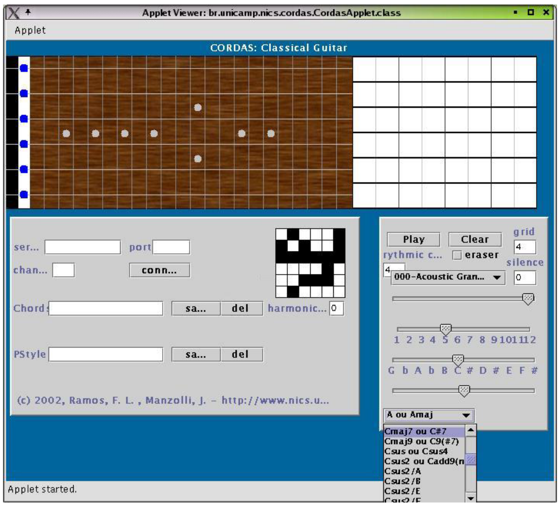

Figure 5 *Cordas* interface

4.2.2 From Play Style to Chord Orbits

To control chord sequences in real-time, we developed a mathematical model to describe the co-ordination between the left and right hand of the instrumentalist. This model is briefly presented below [Manzolli 2002B]. As an example, the definitions below are related to matrix $C_{6\times 6}$, describing sequences of chords played in a classical guitar, where the $n = number\ of\ \ string\ \ group$ = 6, each group has one string.

Def.1: We defined a *Co-ordination Matrix*, denoted as $C_{n\times n}$, with n= number of strings group and its entries are defined in the set $\{0,1\}$.

Def.2: The *Tablature Vector* denoted as $\vec{x}$, is an array containing the fret number played on the i^th^ string denoted as $x_i = fret\ number$ $(1 \le i \le n$ with $n = number\ of\ \ strings$).

$$\vec{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6] \qquad \text{(Eq. 1)}$$

In the classical guitar the highest pitched string **E3** is related to x_1, and the lowest pitched string **E1** is related to x_6 with entries $x_i \in \{-1, 0, 1, \dots, m\}$, where $m = number\ of\ \ frets$ and it depends upon the instrument we are using, for classical guitar *m=19*. The entry "-1" is used to mean "*the string is not played*" and the entry "0" is used to mean an *"opened string"*.

Def.3: Given an initial *Tablature Vector* and a *Co-ordination Matrix*, we call $\vec{x}_{K+1}$ the next *Tablature Vector* in a sequence $\{\vec{x}_0, \vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$. This sequence is called here as *Chord Orbit* and it is generated by the following operation:

$$\vec{x}_{K+1} = C \cdot \vec{x}_K \qquad \text{(Eq. 2)}$$

where $\vec{x}_K$ *is the* k^{nd} *Tablature Vector and* ***C*** *is the Co-ordination Matrix.*

In order to confine the entries of the $\vec{x}_{k+1}$ in the set $\{0, 1, \dots, m\}$ we apply at it *k-step* the following operation:

$$\vec{x}_{K+1} = [x_1 \bmod m, x_2 \bmod m, \dots, x_n \bmod m] \quad \text{(Eq. 3)}$$

where the **mod** *operation gives the* Z_m *correspondent for each entry.*

Def. 4: Given a T_i (a value from the MIDI Note Number Table) with $1 \le i \le n$ for each string, a note number y_i played by *Cordas* is defined as:

$$y_i = T_i + x_i, 1 \le i \le n \qquad \text{(Eq. 4)}$$

where $x_i \in \{0,1,\dots,m\}$ *and it is generated by equation (3) or the entry* -1 *is assigned by the user.*

5. Results and Conclusion

The most interesting result was to listen to musicians performing a Jam Session each one using a different DMI. We have tested this situation using the MIDI Server and the resultant sound was really interesting. It was a complex interweaving of sonic trajectories resembling a chat situation where users write sonic messages represented by MIDI events. Each of the systems presented here was explored in different situations and we comment below their performances.

5.1 *Rabisco* Soundscapes

The *Rabisco* applet application is currently used in the PGL project (Partnership in Global Learning). *Rabisco* proved to be a fun way to stimulate children to enhance their sound perception and musical skills, see http://www.nics.unicamp.br/rabisco/. This application is used by children between 8 to 10 years old in fundamental schools of USA, Mexico and Brazil.

The *Rabisco* application version was used in the *ADA* project. *Rabisco* produced the sound patterns controlled by a large neural network system. The patterns designed with *Rabisco* were applied to the soundscape (http://www.ada-exhibition.ch) presented during six months and over of 550 thousand visitors have interacted with this installation at the Swiss National Exhibition Expo.02.

5.2 *Cordas* Applet Instruments

As mentioned above, the classical guitar was the first instrument studied, but *Cordas* allows an easy creation of new instrument instances or applet instruments. We implemented a series of applets such as the Chinese *Pipa*, Andino *Charango* and the African *Kora* among others and they can be played in the official *Cordas* Home Page (http://www.nics.unicamp.br/cordas/) The applets showed the potential of the software and its flexibility resulting of the usage of class abstraction and *inheritance*. The mathematical model used to define the Co-ordination Matrix is simple enough to explored *Cordas* as a dynamic system and several generative algorithms can be used to control chord orbits.

Acknowledgments

We thank the Brazilian Research Agencies CNPq and Fapesp for the financial support to Jônatas Manzolli. Fernando Lindner Ramos and Márcio de Oliveira Costa were supported by Faep/Unicamp scholarships. We also thank Márcio Saragiotto for implementing the Cordas real-time MIDI dumping and making the last editing of the paper.

7. References

Bischoff, John., R. Gold, and J. Horton. 1985. "Music for an Interactive Network of Computers." In C. Roads. and J. Strawn, eds., The Foundations of Computer Music, Cambridge, Massachusetts : The MIT Press. pp. 588 – 600.

Brown, Chris. 2000. “Talking Drum**:** A Local Area Network Music Installation.**”** In the Leonardo Music Journal, Cambridge, Massachusetts : The MIT Press. Volume 9, pg: 23-28.

Burk, Philip L. 2000. Jammin'on the Web - a new Client/Server Architecture for Multi-User Musical Performance.Visual. In Proceedings of ICMC 2000, Berlin, Ed. The International Computer Music Association, 117-120.

Costa, M., Manzolli, J. & Sharoni, D., 2002. "A Distributed Interactive Composition Tool", SIGGRAPH, San Antonio, USA, pg: 298.

Costa, M. O. & Manzolli, J. 2001. Toolbox para Aplicações Musicais na Internet. In Proceedings of SBC&M 2001, Recife, Brazil, pg: 125-128.

Helmuth, M., 2000. Sound Exchange and Performance on Internet2. In Proceedings of ICMC 2000, Berlin, Ed. The International Computer Music Association, 121-124.

Hwang, J. & Kim, G. J., 2000. Design and Analysis of Virtual Musical Environments (VME). In Proceedings of ICMC 2000, Berlin, Ed. The International Computer Music Association, 125-128.

Jordá, Sergi. 2000. “Faust Music On Line (FMOL): An Approach to Real-time Collective Composition on the Internet**”** In the Leonardo Music Journal, Cambridge, Massachusetts : The MIT Press. Volume 9, pg: 5-12.

Manzolli, J., Costa, M. O., Ramos, F. L., Fornari, J. E. & Sharoni, D. 2002A "Solutions for Distributed Musical Instruments on the Web". In Recent Advances in Java Technology, eds. Power, James & Waldron, John. Computer Science Press, Trinity College, Dublin. pg: 117-125.

Manzolli, J., Ramos, F. L. & Carrilho, F. 2002B. "Constructing Distributed Fretted Instruments for the Web". Proceedings of the Generative Arts Conference, Politecnico di Milano University, Milan, Italy, pg:19.1-19.9.