Touches on the line:
Sharing Csound scores using web server and mobile phones

Antonio Deusany de Carvalho Junior[1]

[1]Instituto de Matemática e Estatística
Universidade de São Paulo, São Paulo,Brasil

dj@ime.usp.br

Abstract. *This paper presents the results of an experiment which was based on an application developed for Android containing Csound embedded and sharing scores among multiple users during a distributed musical performance via the Internet using a web application in Ruby on Rails and hosted as web server. Results detail are described in order to present new perspectives for developing applications aiming to collaboration on mobile performances.*

1. Introdução

A variety of digital content is distributed through the Internet every second in order to provide interaction opportunities between people and contents in a highly collaborative reality [Mather et al. 2009]. The flow of content has increased so much that lots of content manager systems have been developed to make faster editing and delivery of content between users [Brampton 2008]. Multimedia content can also be available using these systems and increase the possibilities of the interaction practices[Chapman and Chapman 2000].

In this paper, we discuss technologies related to data communication over the Internet, focusing on multimedia data, or Csound scores, specifically. We also present results from test using an application developed to exchange scores between mobile phones connected to a web server . The test results will be discussed with the aim of promoting the use of such technologies for the development of other ideas involving content manager systems and musical interaction with multimedia technologies.

2. Technologies for content delivering

There are several ways to exchange data and make content available on the Internet. The most common technology used on this case is the web server, nowadays. The development of web server have been intensified over time so that new methodologies as Rapid Application Development (RAD) began to be practiced with the use of tools and specific frameworks. One of the technologies that emerged following the RAD methodology was Ruby on Rails (RoR)[1]. Rails is an open source framework for web applications that supports Ruby language.

RoR provides different ways for user's communication with the web server, among which some specific format of message can be exchanged, and here we have chosen JSON format as the preference. JSON is an open standard for exchanging data that is

[1] http://rubyonrails.org/

based on a text format understandable by humans without technical knowledge of computer science. The format is derived from the Javascript language and presents itself as an alternative to XML format. Sharing content with web server using this format comes to be advantageous also with the ease of encoding and decoding messages.

Web Server and content delivering have been heavily used by diverse kind of application, with special focus on mobile applications. Among mobile technologies, Android operational system has become widespread and several companies have adopted it into their devices. It means that a software developed for Android devices will have fewer restrictions for running whether you have a Sony or Samsung mobile device even from different versions of Android, despite distinct hardware specification used by both companies.

Those technologies are generically disposed to any kind of content, and its use in the music can open several possibilities for sharing content and predispose infinite ways of musical interaction [Iazzetta 2009, p. 123]. Facilitating its use by musicians may not be so simple, but we are going to present some examples in the next section that can illustrate possibilities of integration between new technologies with Csound and music expression.

3. Integrating technologies for sharing Csound scores

There are various possibilities that arise from individual use of each technology mentioned and also from their interactions. In this paper, an easy example is presented using all of them in conjunction with Csound. Both experimental music and non-realtime applications can benefit from user's interaction focusing on ubiquitous and pervasive computing concept. Bearing all of this in mind, a web application had been developed in a RoR web server, and also an Android application that uses that web application as principal way of communication.

Creating an RoR web application to respond to POST and GET web requests requires some commands described below:

```
$ rails new csoundwebapp
$ cd csoundwebapp
$ rails generate model Cscore user:string score:text
$ rails generate controller cscores index show new create
```

After these commands the application is created but some code needs to be edited on the RoR controller file in order the have the requests answered correclty.

File: */csoundwebapp/app/controllers/cscores_controller.rb*

```

class CscoresController < ApplicationController

  rescue_from ActiveRecord::RecordNotFound, :with => :record_not_found

  def record_not_found
    @cscore = Cscore.new();

    respond_to do |format|
      format.html # index.html.erb
      format.json { render :json => @cscore }
    end
  end

  def index
    @cscores = Cscore.order(’id␣desc’).all
    @cscore = Cscore.order(’id␣desc’).first

    respond_to do |format|
      format.html #index.html.erb
      format.json {render :json => @cscores }
    end
  end

  def show
    @cscore = Cscore.find(params[:id])

   respond_to do |format|
      format.html # index.html.erb
      format.json { render :json => @cscore }
    end
  end

  def new
    @cscore = Cscore.new

    respond_to do |format|
      format.html #new.html.erb
      format.json { render :json => @cscore }
    end
  end

  def create
    @cscore = Cscore.new(params[:cscore])

    respond_to do |format|
      if @cscore.save
        format.json { render :json => @cscore,
                      :status => :created, :location => @cscore }
      else
        format.html { render :action => ”new” }
        format.json { render :json => @cscore.errors,
                      :status => :unprocessable_entity }
      end
    end
  end

end
```

The last commands needed are used to create the database where the scores will be saved and also to start the web app with the new configurations:

```
$rake db:migrate RAILS_ENV=”production”$
$touch tmp/restart.txt$
```

The web application is completed and now the web server can be used. Considering the requests that can be sent to the web server, we have to code the POST in JSON format following the pattern:

{*“cscore”*=>{*“user”*=>*“username”*,*“score”*=>*“shared_csound_score”*}}

All request sent through POST are saved on databased for future GET request that will receive the same message as the score sent in JSON format. Supposing the web server address as ‘*http://www.server.web/*”, the GET request for hundredth score needs to be sent to:

http://www.server.web/cscores/100.json

The Android application developed to exchange scores through the web server was based on Csound Android Examples developed by Victor Lazzarini and Steven Yi [2]. The MultiTouchXY application had been modified to play the user’s score created by touch interaction and also send the same score to web server using JSON format. While that, the application send GET requests to web server aiming to receive new scores from other users playing at the same time. The last score sent to web server can be obtained from *cscore controller* using an specific request:

http://www.server.web/cscores.json

4. Tests and results

Tests were made using various specifications In order to evaluate the developed application. The web application was installed on a local server on a notebook, and also on a hosted server in the *DreamHost*[3]. The web server used was *Phusion Passenger*[4], which is free and was developed to easily integrate with Apache servers. Furthermore, this web server is recommended by RoR community as the best way to run web applications developed using RoR [**?**]. The devices used to test the application were *Sony Xperia Play* and *Tablet Multilaser Vibe NB026*.

The communication performance in both scenarios were quite different and we’ve got good results. On local server, the request latency had a variation from 5 to 500ms, while on hosted server the average latency for processing requests was form 100 to 1000ms. The main problem on local server was due to wireless half duplex connection specification, so the more devices sharing the same wifi connection the more latency we get. On the other hand, the hosted server permits each device to use different wifi connection to share the scores. Even with higher latency, the users can be connected from anywhere in the world and so the experimental performance can get valuable aesthetic characteristics.

5. Conclusions and remarks

As Android devices are becoming more available, the use of mobile applications during performances also become more practical, and the interaction with audience can easily be solved using such technology. The combination of several technologies can bring different perspectives to provide new applications previously unimaginable. Using mobile

[2] `http://sourceforge.net/projects/csound/files/csound5/Android/`

[3] http://www.dreamhost.com/

[4] https://www.phusionpassenger.com/

devices connected to the Internet to communicate through a web application on a web server can bring many solutions for problems that were difficult to be solved before.

The tests performed in this work put together some technologies in order to promote musical interaction between users in a ubiquitous manner in cases that there isn't dire need exchange of information in real time. The tests brought important results related to latency. Delays in communication have been found relatively low, while the worst problem came up to be on the local server environment, considering that wifi connection isn't full duplex as wired connection. So the best scenario for using this application model will depend on how many devices will share scores at the same time.

This work predispose a basic overview of what can be done to provide the use of free technologies for collaboration and interaction during performances using mobile devices. It is expected that several ideas and projects may arise from the solutions presented in this paper. Using all those technologies and putting them together in a harmonic way can benefit from user's interaction to monitoring systems, and considering that technological evolution tends not to stop, many other combinations are likely to be made ??in favor of such practices in the future.

References

Brampton, M. (2008). *PHP 5 CMS Framework Development*. Packt Publishing Ltd.

Chapman, N. and Chapman, J. (2000). *Digital Multimedia*. John Wiley & Sons.

Iazzetta, F. (2009). *Música e Mediação Tecnológica*. Perspectiva.

Mather, T., Kumaraswamy, S., and Latif, S. (2009). *Cloud Security and Privacy ? An Enterprise Perspective on Risks and Compliance*. O'Reilly.