VOL. 14, NO. 2, JANUARY 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

TOWARDS AN ONLINE MUSICAL OBJECTS REPOSITORY FOR ONLINE CUSTOMIZATION AND REPRODUCTION OF AUDIO FILES

Diego Fernando Rojas[1], Yony F. Ceballos[1] and German Sánchez-Torres[2]
[1]Faculty of Engineering, Universidad de Antioquia, Medellín, Antioquia, Colombia
[2]Faculty of Engineering, Universidad del Magdalena, Carrera, Santa Marta, Magdalena, Colombia
E-Mail: yony.ceballos@udea.edu.co

ABSTRACT

The musical objects repository is a web application created to orderly store musical files, cataloging them using configurable metadata for easy reference. It also allows the reproduction of online files through playlists, as well as search and queuing. It was built using technologies such as JQuery to handle the client side, PHP for the server side, AJAX for communication and MySQL for data storage. The designed interface was designed to allow navigation without interrupting playback. The forms, both the creation of objects and the search, have aids and validations to avoid repeating files, minimize the entry of erroneous data and facilitate queries. The use of the repository is efficient for specific purposes, and it is easy to create, to include content, create users and user groups.

Keywords: objects, metadata, music, streaming, wiki.

1. INTRODUCTION

During the last century, radio was the most relevant technology for transmission of music. However, thanks to the massification of internet and the technological improvements in the network, music is now easily available through the web [1], [2]. Nonetheless, information on the internet is not consolidated: repeated, incorrect data is a widespread occurrence [3].

The musical object repository is a tool we aim to turn into a reliable, succinct information source for allowing users to query and play musical files. Most importantly, it will allow them to manage content - *e.g.,* by uploading files and tagging them with metadata - and validating the quality of the uploaded data.

The querying process becomes easier if files and information are consolidated in an organized and categorized way. Furthermore, the process becomes efficient and reliable if the verification and updating tasks are shared among several people.

Since internet evolved into what is now known as Web 2.0, users became a key party in content creation. They do this by making use of a great variety of publishing tools such as blogs, wikis and social networks [4], [5].

Today there are many applications for storing and sharing audio-visual content, albeit their goals are different from those of the musical object repository (MOR). Sites such as Youtube and the defuncts Musicuo and Grooveshark are centered on the multimedia files reproduction, whereas the focus of the MOB is to organize information.

The repository does allow users to play files and introduces improvements concerning the sites mentioned previously. It implements features for making navigation and reproduction easier, such as server-client communication using AJAX [6]. This technology allows communication to be carried out asynchronously, enabling users to carry out navigation and reproduction simultaneously.

Previously, music was only transmitted through radio stations. These stations were the ones to choose the genres and artists to be diffused, for exploiting the most popular ones. The business of radio stations consists of reaching the highest number of listeners. Given that there are many kinds of listeners, there exist as many radio stations. This turned radio into one of the most popular entertainment forms [7].

However, in the case of Colombia, music transmitted by radio stations comes mainly from within the country. Artists from Spain, North America and the rest of Latin America and Brazil produce the bulk of music that can be listened through Colombian radio stations. A limited amount of the music transmitted by these radio stations comes from elsewhere in the world.

Thanks to the internet we now have access to any kind of music from anywhere in the world. In the latest years, artists and groups from all around the world have become known in Colombia, while new genres and languages gain popularity. Music is obtained through the internet in the form of files, which are downloaded and kept on personal computers or mobile devices [8]. As the number of files stored becomes exceedingly big, information becomes impossible to complete, manage or update. This is made worse by the fact most of them do not include basic metadata, *e.g.,* the names of the files and artists.

Music pieces whose artists or names are not known can be found in many personal collections. Repeated files are common, and there are even cases where files disappear without a trace. Sometimes, music belonging to the same artist cannot be grouped due to his or her information having been written incorrectly. Everything can be traced back to disorder within information. Existent systems for uploading and reproducing music do not focus on information management. Thus, information is disorganized, repeated, inconsistent, and cannot be correctly consolidated.

2. STATE OF THE ART

An in-depth look will be given to the evolution of the internet and web services during the last years. This is done for analyzing how pertinent a meta-categorized musical object repository is, and for showing how users take an increasingly important role in building up content on the web.

Companies belonging to the informatics sector such as O'Reilly and MediaLive International studied in the year 2004 the concept of Web 2.0. O'Reilly defines it as follows:

Web 2.0 applications are those that make use of the advantages of the web. They offer a service that is constantly updated and improves as more people use it. These applications use and mix data from multiple sources, including individual users, at the same time as they offer their data and services for others to reuse. This creates a networked 'participative architecture' which goes far beyond the Web 1.0 page to offer richer user experiences [9].

The main feature of the Web 2.0 is the widespread abandonment of static content, which previously was read-only, for making the user a key part of its creation. This makes it possible for anyone to upload their own opinions, comments, media and even web pages about whichever topic they desire. [5].

There are practical examples of user participation in content creation. From small contributions in forums or wikis to blog and full pages shared in social networks and other services that have made internet become the main tool for academia and business [10].

The main reason for the success of Web 2.0 services is that they make use of collective intelligence. This concept is built upon the idea that several dozens, hundreds or thousands of individuals can pool their ideas together through debate and discussion for obtaining a more accurate and reliable consolidate of information [5].

A. Relevant projects

Next, we show some of the most popular Web 2.0 tools. These illustrate the concept of collaborative content creation, through a community which constantly integrates content.

- Wikipedia

A free encyclopedia based on allowing any user to include or edit any content. It is a huge vote of confidence that applies Eric Raymond's maxim [11]: "with enough eyes, any error is superficial". Wikipedia is often among the first results of the query results of Google. It has become one of the most searched portals on the internet due to the diversity of its contents, which are presented in a simple, easy-to-read format.

- YouTube

A web site where users can upload, share and visualize videos. Currently, it belongs to Google and uses an Adobe Flash-based online player to serve content to users. It is one of the most popular pages of the web due to the amount of content it provides. It has quickly become a pillar of electronic culture [12]. At first, Google tried to compete with YouTube with its video sharing system, Google Video. However, it was unable to surpass the popularity of YouTube and decided to buy it in 2005.

- Yahoo! Answers

An innovative question/answer system where any question can be formulated and any user can provide a reply. This allows users to share knowledge, opinions, and experiences. Afterward, answers are reviewed and put under community scrutiny to select the most appropriate reply. This service is free of charge and was launched in 2005 [13]. Despite retaining popularity in some countries, it has been displaced by similar services such as Quora in the English-speaking sphere.

- Grooveshark

A free portal created for sharing and listening online to the music of all kinds. It allowed the user to upload and search songs, to create playlists and to share them with friends. Its interface was simple and configurable [14]. It was successful and enjoyed a sizeable community, but it ended up taken down in 2015 after a copyright claim by Universal Music Group, Sony Music, and Warner Music Group.

- Musicuo.

A system inspired by Grooveshark whose main feature was it was built fully on HTML 5, which supports audio reproduction from the browser without the need to install plugins such as Flash. This project was considered the Spanish alternative to Grooveshark [15], but it was abandoned by its creators in fear of a copyright claim.

Despite the popularity of these tools, none of them is flexible enough for allowing users to organize and share metadata in collaboration with others.

3. METHODOLOGY

This solution is based on two important concepts shown to be successful in the context of the Web 2.0: wiki and streaming.

The wiki concept consists of the construction and edition of content by multiple users [16]. Sometimes, contents are published under certain restrictions or after being subject to scrutiny [17]. By implementing the concept of a wiki, it is expected users include content in the MOR and controls any errors that might arise. In this way, they will not only change the information belonging to their objects but also of those uploaded by other users, if they have the required privileges.

The streaming concept is used for online multimedia files reproduction. Using streaming, it is hoped that users will be able to enjoy contents stored in the MOB from their browsers.

A. Design patterns

Design patterns are proven solutions to recurring problems. They define a methodology we can reuse in

different projects for reducing programming time by not having to start from the ground up.

- Model-View-Controller Pattern (MVC)

The MVC pattern seeks to divide the application in three basic layers [18]:

- View. The presentation layer where events and petitions occur. The user interacts with this layer.
- Controller. The layer that directs application flow. It receives user petitions, processes information and outputs a view or a message to the user.
- Model. The layer that interacts with persistent data base information. It is invoked by the controller when necessary.

- ActiveRecord Pattern

A pattern employed for data access, which maps tables or rows resulting from a query into class objects for easily manipulating data from the tables through methods [19]. This pattern is usually employed by persistence and relational-object mapping tools.

It was proposed to build a web site specialized in systematically and orderly storing multimedia files through metadata. In this scheme, the users would be the ones in charge of including, managing and controlling the contents of the application.

A problem was identified concerning musical file querying. It was stablished that the lack of metadata control was a possible cause, and a hypothesis was put forward: a platform can be created where users can collectively provide enough metadata to remove information disorganization.

To validate the hypothesis, it was proposed to develop a web system, and several objectives were sketched for illustrating what was to be accomplished. A preliminary study regarding similar systems was carried out, and their advantages and disadvantages were evaluated to determine aspects to consider and improve. We delimited the objectives and defined a development timetable.

Once the proposal was approved, the first step was to define the tools we would use, and the data model that would support the requirements of the application. Afterward, we drew a class diagram and defined roles and use cases. Then we began the development of the MOB. Several frameworks, plugins, libraries and design patterns were employed for this purpose.

Components such as login, advanced search, object creation and the suggestion provider were created one by one. Afterward, the playlist functionality was created and integrated with the employed web player. The local development site was constantly synchronized with an online test server for carrying out synchronization tests between servers.

B. Requirements

The system had to fulfill the following characteristics:

• Functional Requirements
- Navigation within should not limit the reproduction of the files.
- Differentiate user types according to roles.
- It must allow users to create objects and upload files
- It must permit the search of objects through its metadata according to a configuration-based setup
- It must allow searching for objects within the playlist.
- It should allow playing online files through streaming.
- Multiple options to control the file reproduction order, including the option to queue up files.
- Playlists must be able to be stored for future use.
- It must allow suggestions on objects created by other users to correct errors and complete missing metadata.
- It must permit to accept and reject user suggestions.
- It must allow metadata configuration to be adaptable.
- It must allow the creation of new metadata.

C. Employed tools

MOB is a web application based on CodeIgniter, a PHP framework which controls interaction with the web server and the database server. jQuery is used on the client side. It is a library which streamlines interaction between the user and the browser. jQuery is also charged with managing communication between server and client, through the AJAX technology.

- PHP

It is an interpreted programming language used to create dynamic web pages. It is open source, multiplatform and facilitates connection with databases such as MySQL. It also allows applying object-oriented programming techniques [20].

- CodeIgniter

It's a PHP application development framework that allows using patterns such as MVC [21].

- MySQL

A relational database management system. Its features include being multiplatform, multithreaded, multiuser and open-source [22].

- jQuery

A JavaScript framework that eases handling user-generated events, and supports a wide variety of plugins [23]. It allows JavaScript code to be simplified, to chain functions, to interact with easily and modify DOM tree

objects, to manipulate CSS styles and to generate effects and animations. Furthermore, it allows communication between the client and the server to be carried out asynchronously through AJAX functionalities [6].

- AJAX

Asynchronous JavaScript and XML. It is a web development technology that is executed on the client side and maintains background asynchronous communication with the server. This way petitions can be done without reloading the whole page. JavaScript is the language where the AJAX functions are executed, whereas the data access is done through the XMLHttpRequest object, available in most modern browsers [6].

- jQuery-UI

jQuery User Interface is a library that extends the jQuery basic functions through several plugins, facilitating the user-defined components creation [24]. Some of these components that are used in the MOR are:

- Autocomplete: Allows displaying a list of options as a suggestion for a text field. Used to show the suggestions when filling the forms of search and creation of metadata objects that are "automatic text" type.
- Datepicker: Allows showing a calendar. Used for "date" type metadata.
- Dialog: allows to display a dialog box. Used for important messages that cannot be overlooked.
- Resizable: Allows you to change the size of a container by mpouse dragging. Used to expand or reduce the playlist.

- Juploader

It is a jQuery plugin that allows file upload to be carried out asynchronously. The extension used iFrame to emulate the AJAX technique and was developed by Michel Manchego. Among its advantages are that it allows file upload to be limited depending on their extensions, it does not require the page to be reloaded, it's easy to implement, and it allows events to be controlled. However, when the file is selected, it must be sent immediately to the server [25].

- jPlayer

It is a jQuery plugin that allows a music player to be embedded into the web page, using HTML5 and JavaScript technology. However, there is also a Flash option that can be used when required. The HTML5 version supports mp3, mp4 (AAC/H.264), ogg (Vorbis, Theora), webm (Vorbis/WP8) and wav, among other formats [26].

- getID3

A PHP class that allows audio file properties to be extracted, including title, author, length, lyrics, etc. [27]. For the MOB, we only need to extract the length property from songs.

D. Development

Three of the previously mentioned tools were used as a reference for the development of the MOB: YouTube, Grooveshark, and Musicuo. This was done to analyze their main features, advantages and possible improvements for each one. This information is summarized in Table-1.

Table-1. Similar systems: favorable and unfavorable features.

System	Favorable features	Unfavorable features
YouTube	- Related video recommendation - Stores playlists - Allows queueing videos	- Videos may be difficult to find - Does not allow simultaneous reproduction and navigation: navigation impedes reproduction
Grooveshark	- Allows reproduction and search to be carried out simultaneously - Includes radio stations	- No advanced search - No song queuing - Important metadata such as song lyrics can't be stored
Musicuo	- Has a dynamic interface - Works without the Adobe Flash player plugin - Stores search history	- No advanced search - No song queuing

One of the most important objectives is to allow navigation and reproduction to be carried out simultaneously, with the lowest number of interruptions. We employ the AJAX technology for achieving this since in most cases file reproduction is interrupted when changing pages or reloading them.

Thanks to AJAX, petitions to the server can be done asynchronously. This way, sections of the application can be loaded without reloading the whole page. This is how reproduction is kept running at the same time as navigation is done.

Another goal is to avoid file redundancy and errors when creating new objects. The metadata input form has two features for helping avoid user error.

- Autocomplete

For each metadata, an "Autocomplete"-type help is shown for keeping the user from making orthographical

errors and validates whether the inputted metadata is already in the database.

- Active search

It is an advanced search that is done at the same time as the users inputs each metadata one by one. This is done for showing users coincidences with objects that exist. It is expected that if the user finds that the song they wish to upload already exists, they will desist as to avoid having repeated files.

Additionally, two features were implemented to improve playlist management.

- Search within playlist

This functionality allows objects belonging to a playlist to be easily located, through a search field placed on the top of it which executes the search as it is written, finding coincidences within the playlist and hiding objects which do not coincide.

- Queueing objects

Queuing objects allows the order of the playlist to be defined by the user, signaling which songs they wish to play next.

Application

E. Program flow

The internal CodeIgniter procedure for managing petitions is detailed in Figure-1. The user interacts with the browser directly through events, which are controlled by jQuery. A petition is carried out using AJAX to the address of the controller charged with continuing the process of handling the request of the user.

The controller can employ libraries and models according to the needs of each request. Libraries store business classes and the model is the path from which the database is accessed. Once the answer to the petition is received, there are two ways to communicate it to the browser. Through a message or by loading a view. If the response is a view, jQuery shows it on the page. If it is a message, it carries out the adequate action depending on the message. The workflow within the MOB is shown in Figure-2.

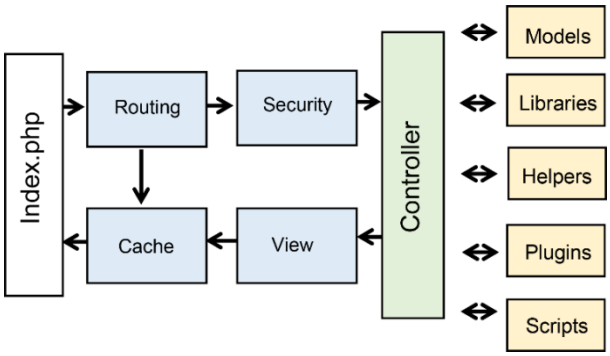

Figure-1. Procedure carried out by the CodeIgniter framework to handle a request.

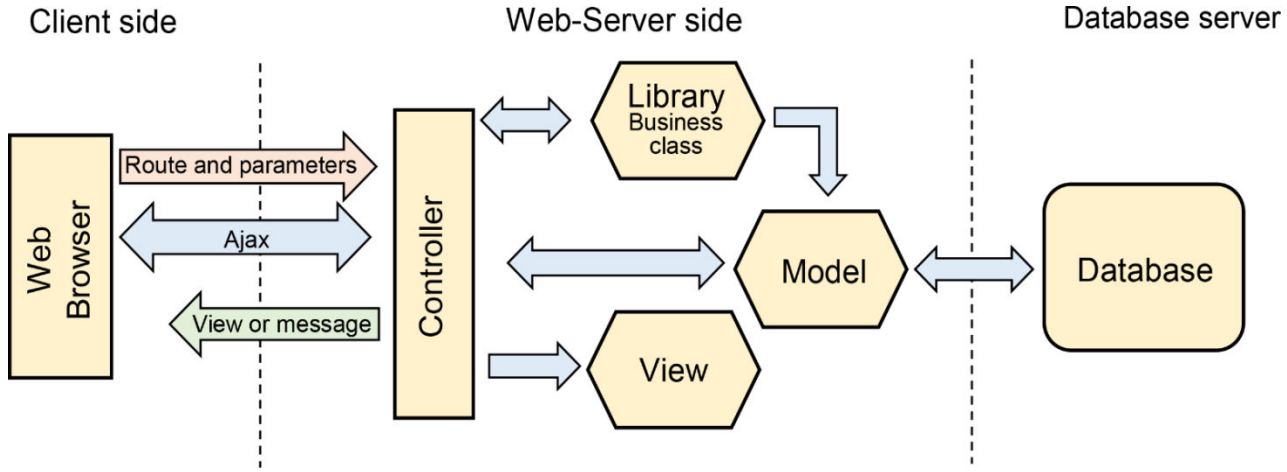

Figure-2. MOB's flow, with the client, web server and database server sides as well as the interactions between them.

F. Presenting the application

The 1.0 version of the MOB is presented as a prototype with the main functionalities, which allow it to accomplish the objectives proposed for its development.

- Object creation
- Object search
- Error checker
- Object reproduction
- Creation, control and storage of playlists
- Configurable metadata.

Figure-3 shows the main screen of the MOB application.

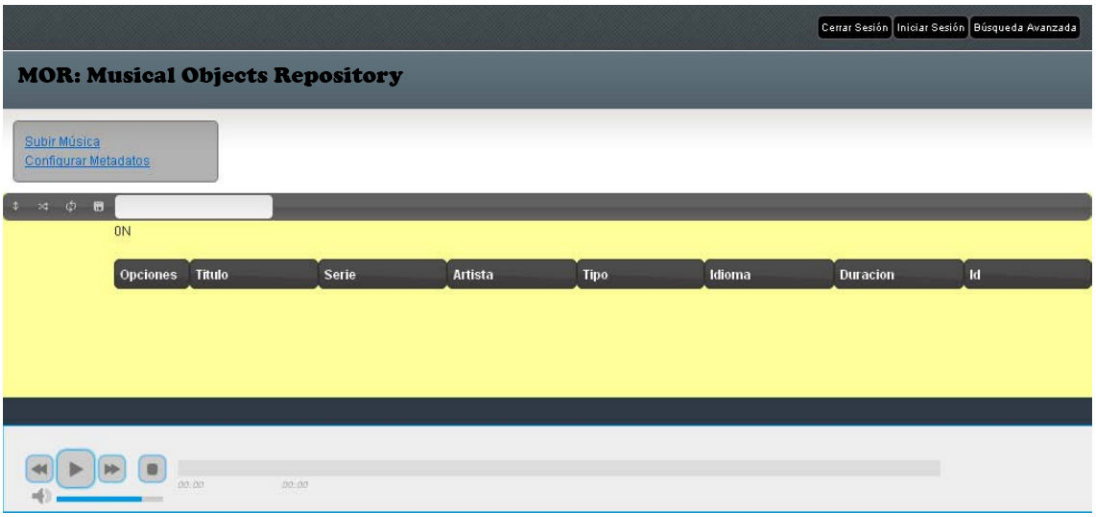

Figure-3. MOB's main view.

G. Metadata configuration

We make metadata configurable for providing freedom over the repository type that the administrator wishes to implement, based on the data they consider important. Each metadata has the following options:

- Visible

Indicates whether the metadata is shown in a situation. It can take the following values, separated by a hyphen (-):

- U: Visible in the object creation form.
- S: Visible in the object recommendation form.
- B: Visible in the object search form.
- R: Visible in the search results form.
- L: Visible in the playlist.

- Enabled

Indicates whether the field is used.

- Mandatory

Indicates if the metadata is mandatory or not in the object creation form.

- Default

Indicates a default value, for the cases where the user does not input any data.

- Order

Shows the order in which metadata are displayed in forms.

- Field type

Indicates how the field is to be shown in forms and the way they must be validated. It can take the following values:

- Text: Displays a normal text field, without validations.
- Automatic Text: Shows a text field which is validated through recommendations for preventing data from being inputted erroneously.
- List: Displays a list with several options to choose from, ideal for metadata with few possible values.
- Text area: Displays a text area
- Date: Displays a calendar.

- Strict

Field types such as "Automatic Text" and "List" can be defined as strict or not for determining how they are validated. If the field is strict, values that are not present in the database cannot be inputted. Otherwise, a recommendation is shown, and whether it is accepted or not, the value is stored.

Due to the fact metadata are configurable, forms such as the object creation form, the advanced search form and the recommendation form are loaded dynamically according to the configuration. The configuration is extracted from the database and saved into the current session. It is key to pay attention to this point since the metadata configuration can be changed, but it will not be reflected until the current session is cleaned and the configuration is reloaded from the database. This is done to improve performance and avoid queries with little or no variation.

To add new metadata, the name, which cannot be repeated, is inputted, along the relevant configuration data. Attention must be paid to the fact that whether the metadata is created as strict or not, once it is created, this property cannot be changed. That's to say, strict metadata cannot be converted into non-strict metadata. This limitation arises from strict metadata having a different database structure to that of non-strict metadata. It is important too to signal that once the metadata is saved, a line must be added to the language file.

H. Object creation

To create objects, the "Upload Music" button is pressed. The file select button appears, and once the file is selected, a temporal copy is sent to the server. If this

process is successful, a form appears and allows the object's metadata to be inserted.

Strict metadata have two presentations for allowing data unicity to be preserved. In the form of a list (used for metadata which has few options) and as autocomplete-aided text fields.

When metadata is strict, the inserted value is immediately compared with those in the database to determine whether it is valid or not. If it is not, the user can input a new value for this metadata.

As metadata values are inputted, an object search is carried out to avoid object duplication. Once the form has been filled with all the mandatory values, and the strict metadata contain valid values, the information can be sent to the server where the object is created. Finally, the file that was previously uploaded is associated with it and transferred to a new location.

J. Object search

Advanced object search allows queries to be carried out based on any of the configured metadata. Afterward, contents can be added to the playlist.

K. Creation, control, and storage of playlists

The playlist includes its search form, which allows objects to be searched within it. The list can be used in one of three modes: normal reproduction, random reproduction and repeat song. The list cannot be ordered, but the next song can be chosen through the queuing option. This allows the order of the list to be modified. Figure 4 shows the list of options available to manipulate the playlist. It includes changing the size of the list, randomizing it and doing a search in the text field.

Figure-4. MOB playlist options.

I. Difficulties

Due to the use of the jUploader plugin, the file must be sent immediately to the server as it is selected. Thus, the creation of the objects must be carried out in two phases. First, the file is uploaded, and then, the data is inputted. When both tasks have been completed correctly, the file is stored in the database, and it is possible to make use of it. If any of the two tasks fail, the whole procedure must be done again.

File upload itself is done in two steps. First, the file is uploaded to a temporary folder, and when the object is created it is sent to its final location, and the temporary file is deleted. However, if the process is not completed, it remains stored in the server, and the administrator must periodically delete content from the temporary folder. Furthermore, during testing, it has been discovered that the jUploader plugin sometimes fails, preventing other files from being uploaded until the whole page is reloaded. Sometimes, a song can be played up to two or three times despite the list being in "random" mode.

On the other side, when metadata is inserted, it cannot be deleted. The only solution is to disable it. Additionally, when metadata are added, the language file must be opened manually to add its key and the value that is desired to be shown. Forms that depend on metadata are created dynamically. However, existing metadata names cannot be modified.

On top of solving the technical problems that were exposed, if the MOB proposal is to be published massively, several floating tooltips and helps will need to be added to make the system less invasive. It would also be necessary to allow user registration, to implement a scoring system and to design a privilege-based system to provide the tool with more security.

As music producers and publishers grow more aggressive in their copyright claims (as shown in the Grooveshark case [14]).

4. CONCLUSIONS

With the strategies implemented by the MOB, the number of repeated musical objects is reduced. However, it is not difficult to input erroneous data. Configurable metadata allow generating versatility in the contexts of implementation, with a high degree of customization.

Indeed, the MOR is most efficient when used for specific purposes by limiting the kind of sound files that can be uploaded, *i.e.* music from a particular genre. Anybody can create their own music files repository using the MOB and a group of users can maintain or enjoy the content.

The MOR satisfies online music files reproduction needs. It streamlines user interaction with search and queuing options. However, there exist technical and usability problems that must be solved if a massive version is to be released to the public. Additionally, if MOR is to be made more accessible, it is important to consider ways to avoid legal obstacles which might arise. One possibility which is being considered is to limit user capability to access files stored in the platform (or to store MD5 hashes), allowing only metadata to be shared instead.

REFERENCES

[1] M. Graziano y L. Rainie. 2012. «The music downloading deluge», Pew Internet Am. Life Proj. Httpwww Pewinternet Orgreports Accessed. Vol. 1.

[2] U. Dolata. 2008. «Das Internet und die Transformation der Musikindustrie». Berl. J. Für Soziol. 18(3): 344-369.

[3] M. A. Keller, «Linked data: A way out of the information chaos and toward the semantic web». Educ. Rev. 46(4): 10-11.

[4] Paula Fernanda Cadena. 2010. «Introducción al uso de la Web 2.0 en el estado Colombiano». Sep-2010.

[5] D.-R. Rodríguez-Palchevich. 2008. «Nuevas tecnologías Web 2.0: Hacia una real democratización de la información y el conocimiento.».

[6] Javier Eguíluz Pérez. «Introducción a AJAX».

[7] US Media Consulting. 2013. «LATIN AMERICA'S MEDIA MARKET 2013».

[8] M. Torrens, P. Hertzog, y J. L. Arcos. 2004. «Visualizing and exploring personal music libraries.». en ISMIR.

[9] T. O'reilly, What is web 2.0. 2005.

[10] B. K. Lee. 2010. «Epidemiologic Research and Web 2.0—the User-driven Web»: Epidemiology. 21(6): 760-763.

[11] E. S. Raymond. 2010. Cathedral and the Bazaar. www.snowballpublishing.com.

[12] L. Siri. 2008. «Un análisis de YouTube como artefacto sociotécnico». Diálogos Comun. Vol. 77.

[13] A. Rechavi y S. Rafaeli. 2012. «Knowledge and social networks in Yahoo! Answers». en System Science (HICSS), 2012 45th Hawaii International Conference on. pp. 781-789.

[14] Panagiota Papakos. 2008. «Grooveshark brings legal music sharing to Gators and the entire world». InsideUF.

[15] V. Pimentel. 2017. «Musicuo es la alternativa española a Grooveshark (¡Y en HTML5!)» Genbeta, 02-dic-2010. [En línea]. Disponible en: https://www.genbeta.com/multimedia/musicuo-es-la-alternativa-espanola-a-grooveshark-y-en-html5. [Accedido: 11-ago-2017].

[16] J. Villarroel. 2007. «Usos didácticos del wiki en educación secundaria». IKASTORRATZA E-Rev. Didáctica. 1: 1-7.

[17] A. Di Iorio, F. Draicchio, F. Vitali, y S. Zacchiroli. 2012. «Constrained wiki: the Wikiway to validating content». Adv. Hum.-Comput. Interact. 2012: 8.

[18] T. Reenskaug. 2003. «The Model-View-Controller (MVC) Its Past and Present Trygve Reenskaug, University of Oslo (trygver@ ifi. uio. no)».

[19] K. Marshall, C. Pytel, y J. Yurek. 2007. «Introducing active record». Act. Rec. Databases Ruby Rails. pp. 1–24.

[20] M. P. Duarte y I. M. Pérez. 2007. Programación en PHP a través de ejemplos. Recuperado el.

[21] «Welcome to CodeIgniter - CodeIgniter 3.1.5 documentation». [En línea]. Disponible en: https://codeigniter.com/user_guide/general/welcome.html. [Accedido: 11-ago-2017].

[22] J. Murach, Murach's MySQL. 2015. 2nd Edition, 2nd edition. Fresno, CA: Mike Murach & Associates.

[23] B. Bibeault y Y. Katz. 2010.jQuery in Action, Second Edition, 2 edition. G reenwich: Manning Publications.

[24] T. J. VanToll, jQuery UI in Action. 2014. 1 edition. Shelter Island, NY: Manning Publications.

[25] Miguel Manchego. «Miguel Manchego - Desarrollo web y marketing digital».

[26] 2014. «jPlayer Developer Guide».

[27] James Heinrich, «getID3()».