ABSTRACT
Along with the growth of digital music consumption, the need for new solutions for browsing and exploring music libraries arises. However, the current main solutions for browsing and searching through music libraries have several limitations, such as: dont use content-based information; dont allow users to understand the collection organization; the creation of playlist remains based on textual metadata. Starting from these limitations, we developed a new solution that combines three main components: information visualization techniques (based on semantic ordered treemaps), music information retrieval techniques and dynamic filtering, to offer users a more efficient, flexible and yet, easy to use solution for browsing music libraries. Experimental evaluation reveals that our application is easy to use, and allows users to perform a more effective and fast navigation, while having a deeper knowledge of their library. The used techniques were considered as a strong value, allowing users to discover music and interactively browse their libraries. Satisfaction inquiries showed users satisfaction on browsing, filtering and creating playlists.

Author Keywords
Music Library Browsing, Information Visualization, Music Information Retrieval, Ordered Treemap, Dynamic Filtering, Audio Snippets

INTRODUCTION
The size of personal music libraries is rapidly increasing as well as the popularity of digital music (e.g. MP3). Due to this, the problem of organizing, exploring and browsing musical collections becomes more clear and obvious. As these libraries increase, new solutions must arise to solve the problems caused by this growth.

Existing solutions present a set of limitations, such as: dont allow users to have an overview of their personal collections, including its composition (so browsing has some limitations as a consequence); dont take advantage of the information contained in music (e.g.: for filtering collections), and finally, the automatic playlists only take advantage of music tags.

Despite being designed to support arbitrary library dimension, currently available solutions to listen to music on our computers present several limitations when subjected to the size of current collections. The main constraints are reflected in the form of navigation and browsing, which is intended to be both simple and effective. Although there are efficient solutions with the ability to create profiles based on user habits and preferences, and several sophisticated engines to search music using semantic information (e.g.: tags attached to tracks such as genre, artist name, etc.), the display of this information has shown little progress or innovation, so that most users do not know their library completely [13]. Additionally the flexibility in playlist creation process must be improved.

In this document we describe a new solution to navigate, explore and view large digital music libraries easily and efficiently. Our solution provides a quick and simple interaction with the music library, and an interface clear enough to allow users to build a visual overview of their collections. Along with this, we created an interactive filtering mechanism based on semantic information, and content-based audio features of music. Using the developed filters, users can simultaneously search for songs that meet their criterion and in an interactive and progressive way explore the collection. Our ultimate goal is to allow users to quickly filter their library, listen to music they like and know, but also discover new music, similar or not, thus increasing the knowledge about their personal libraries.

The solution we describe in this paper, is based on a combination of three key elements: information visualization techniques, music information retrieval\(^1\) (semantic and content-based) and dynamic filtering. Our solution exploits a new visualization technique based on Treemaps [15], to provide a distinct form of interaction with music libraries. Combining this technique with mechanisms of MIR, users can not only browse their libraries using filters based on semantic information, but also through the use of content-based audio features, for example, using similarity between tracks, albums or even artists. Thus, users can achieve a great control over their personal collections, including discover of music with which they were unfamiliar.

\(^1\)From now on we will refer to Music Information Retrieval (MIR) and its abbreviation as the same.
To evaluate the achievement of our goals, we conducted an experimental evaluation. For this evaluation we selected a group of users representing the target user profile for our system, and executed a series of tests with them. In these tests we intended to measure the effectiveness and efficiency of the mechanisms developed in our solution, such as the visualization mechanism or the filtering tools. The test consisted of four parts: introduction to the test, where we described the context and objectives of our work; tasks execution time, having users divided into two groups to obtain more reliable results and study differences between the groups; finally, an informal interview and a satisfaction survey. Results showed that our solution is faster, more efficient, and more effective, providing users a useful and broad overview over music libraries and an innovative form of browsing. Satisfaction results corroborate this fact.

In the remainder of the paper, we present on the next section the related work. After the related work, we present the main findings of a contextual inquiry performed in order to obtain results of users’ habits, behaviours, and needs while listening to music. At the end of this section, we introduce the main design principles that have constrained our solution. After the contextual inquiry section we describe the general architecture of our solution and take a deep look at all of its components. Here we present the communication flow between these components in order to understand our solution behaviour. Next, we describe and analyze the most relevant mechanism of our solution: the interactive filtering mechanism. On the next section, we show the developed prototype to validate our solution. After this, we present the main results from the experimental evaluation and we finish this paper with conclusions about the accomplished work pointing the directions to the future.

**RELATED WORK**

There are two research fields, that by combining the best results from both we could develop better solutions to answer the most pertinent issues identified in the previous section. The two areas under discussion are: Music Information Retrieval (MIR) and Information Visualization.

In recent years, MIR has become one of the most important research fields. Generally speaking, we can describe MIR as an interdisciplinary scientific field involved in the development of automatic and intelligent music processes. The main goal of this area is to allow a set of audio information to be easier to obtain. To achieve this, most of the research in information retrieval has been focused on the automatic music description and evaluation of the proposed methods.

Information Visualization allows us to generate a set of output data from one set of input data. In data output set, information is presented to the user in a way that facilitates the understanding and the readability, depending on the purpose of that visualization technique in particular. There are numerous visualization techniques, which depend on several factors, such as the input the data and the level of detailed information required.

Thus, in the following sections, we analyze the most relevant works of these two areas, showing how they contribute to the creation of more powerful solutions.

**Music Information Retrieval**

There are three basic strategies to solve the problems presented to Music Information Retrieval. The first strategy is based on metadata (information embedded or associated with audio objects), the second approach is based on high-level descriptors of music content, and the last strategy is based on low-level features that can be drawn from musical pieces. We present a brief description of the three identified strategies.

Metadata, also known as semantic information, are the precursors of MIR systems. There are two types of metadata used to describe the music: factual (e.g.: artist name, duration, etc) and cultural (mood, gender, etc). Most current systems use a combination of these categories of metadata. However, there are some problems inherent to metadata: consistence problems, the need of a shared and widely understood vocabulary for tags, and automatic methods for categorize songs.

For all its usefulness, the metadata cannot solve all the tasks of information retrieval of music, due to the complexities highlighted above. This presents an opportunity for the content-based methods. These techniques promise to be able to complement the methods based on metadata giving users access to more music, either through processes of self-discovery, whether by providing a scalable mechanism for browsing the entire digital library.

Features like melody or harmony, represent the so-called high-level features of music content. The high-level information about music, describes the type of knowledge that a listener would have over a sophisticated piece of music. These features are hard to extract although easy to understand, and provide the most intuitive start point for using content-based MIR.

The low-level features describe the audio in the form of acoustic features that are generally well defined and determined on the basis of direct analysis of the audio signal. Although they are well understood and accepted, a drawback of these features is that they don’t say much about the structure of music, as opposed to the characteristics of high-level (to users without technical skills in this subject). Some examples of low-level features are: loudness, spectrum powers, brightness, bandwidth, pitch and cepstrum.

The usage of isolated features cannot be used for complex tasks, such as determining the similarity between two musical pieces. However, based on the results described for the extraction of low and high level features, we can infer about the similarity between tracks. The similarity between music tracks can be done at various levels and with different purposes. The similarity spectrum can go from searching the same audio content, like the works presented in [14, 1], to works like [9, 2], where we only want to search for similar
Information Visualization
Having large digital music libraries is a current and usual situation. However, there is no advantage in having all these songs if we are not able to make an effective navigation and browsing of this collection. In this section we present the most relevant works and their problems, drawing the road to our solution and objectives. The choice of these techniques (and systems) over others, is due to the fact that these present interesting results in relation to navigation, and allow a different and innovative way of structuring and organizing digital libraries.

The Globe of Music, presented in [6] is an alternative solution to browse music collections. Using a spherical SOM [5], trained based on low-level features, visualization and navigation are accomplished through a Geographic Information System (GIS), the NASAWorldWind\(^2\). This system allows users to navigate over a virtual 3D world to explore music collections, using a set of detail layers to aid in the browsing. However, this system has some flaws in navigation, as reported by users after a experimental evaluation: it does not allow users to understand the organization of the library except through hearing the songs, and it does not provide an alternative mechanism to search for music, beyond the traditional text search.

In [11], Pampalk presents a music library exploration system, which uses a metaphor of islands, seas, and mountains. Here, the inherent structure of the music collection is reflected in the arrangement of them. This solution is also based on a SOM [5], trained with content-based audio features for audio similarity (Rhythm Patterns). In Islands of Music, the islands represent musical genres and mountain/valley in the islands represent subgenera. The islands are situated so that similar genres are close and can be connected by a passage, while the different genres are separated by a deep sea. Nevertheless, this solution has some drawbacks: content-based feature extraction mechanism relies on heavy mathematical methods, and has some problems (e.g.: some songs are wrongly classified, slow library processing, etc); has not been tested with a library with significant size (at least 1,000 tracks, as less than 400 songs were used).

In 2005, Frederic Vavrille developed LivePlasma\(^3\) and published it on the Internet. The LivePlasma uses information gathered from Amazon \(^4\) about similarity between items, to create similarity maps between artists (with links between artists to enhance relations). This similarity between items is determined by processing objects that customers buy together with others [8]. For using the system, users only need to enter an artist name, and the application displays a map with similar artists. However, one of the major limitations of collaborative filtering algorithms becomes clear: to receive recommendations, the user has to provide the system with an artist who already knows. For a system which makes recom-

\(^{2}\)http://worldwind.arc.nasa.gov/, accessed on 05/08/2009.

recommendations based on purchasing choices or on what users are currently viewing, this approach makes sense, but on a system where we want to search songs that cannot be easily described, this solution could not make such recommendations.

In [7], Lillie introduced MusicBox, a new solution for organizing music collections into a two dimensional space in a way that rethinks the way people are used to view and organize their collections. This solution explores a set of content-based audio features (such as tempo, timbre, or audio summarization), combined with semantic textual information, to explore the 2D visual representation of the library. In this representation, similar items appear next to each other, and it is possible to distinguish clusters of albums, artists, songs, and genres. Users can then explore these clusters creating intelligent playlists controlled by the available filtering criterion. However, not everything is good in this solution. MusicBox has a complex structure and it is difficult to understand how it works and how the library is organized, especially for ordinary users. Like Islands of Music, MusicBox was tested with a very small music library.

In short, there is still plenty of space for developing new solutions, since some problems still persist.

An important fact is the size of the libraries used in these systems. As the described solutions above showed, there is clearly a gap in viewing music collections of high dimension, in the order of thousands of tracks.

The understanding of the whole collection of music is another aspect that is not fully resolved with the systems described. The solutions described tried to map the music library into a visual representation that facilitates user comprehension about the structure and organization of the library. However, some problems still remain unsolved.

Other aspects that still need improvement to provide better solutions are the browsing and exploration techniques of the collections themselves, accompanied by appropriate filtering mechanisms, flexibility in the automatic playlists creation process, and the improvement and development of metadata and content-based extraction methods.

USER STUDY
Because we are putting the focus on the potential users, we started by performing a study about users habits and behaviours in the organization and exploration of their music collections.

The study was composed of an online survey and of a contextual inquiry. For these, we created a questionnaire divided in three parts, with a distinct goal for each. The first part was designed to characterize users. The second part tried to understand participants habits and behaviours while organizing and exploring music. We tried to figure out how users actually find music, explore their libraries and generate playlists. Finally, in the last part, we tried not only to discover users needs when listening and exploring their libraries, but also
to identify which features they would like to have and what should be the main criteria for querying and browsing libraries and to generate playlists, in a new application for visualization and exploration of music.

We received 127 answers from the online survey, covering a wide range of users. On the other hand, the contextual inquiry was done with ten users, selected to be representative of the potential users. We chose users with collections of around one thousand tracks, who listen to music on the computer very often, and from both genders (six male and four female), with ages between 20 to 40 years old.

Most of users were male (65%); and the majority (81%) was in the interval from 20 to 29 years old. Almost all of them (94%) listen to music on their personal computers; and elected home as the preferred place to do it (84%). On average, 83% of the these participants have a digital music library, the most common size being between 1,000 and 10,000 tracks (40%). There are also some users (17%) who have collections of more than 10,000 songs, which reveal that the majority of users (57%) have libraries with more than 1,000 tracks. However, their collections are not very diverse. Indeed, 71% of the users have rock songs, 61% have pop tracks, while the third place is for electronic music with only 34% (figure 1). From these values, we believe that genre is not a good criterion for structuring personal libraries of music.

Next, we tried to identify what tool users employ to search for songs they would like to listen. Results show that there is no difference between using the specific program to listen to music (47%) and searching directly on the filesystem (45%). From this, we can infer that current applications for music organization do not have any relevant mechanism that distinguishes it from the traditional filesystem structure. Our next step was to understand how users browse their music library. We identified a clear order in the way users navigate through their collections. Forty six percent of users selected Artist as the first criterion to start browsing, 41% selected Album as the second, 41% selected Genre as the third and 53% considered Composer as the fourth criterion. These figures reinforce our previous observation on the use of genre to browse libraries with little diversity. Some participants also referred that they have never browse by Composer (it was only used for classical music). More than 61% of the participants use playlists, being most of them created by selecting individual tracks (60%), 10% using filters and automatic smart playlists and 11% randomly. Another interesting fact is that 57% of the participants do not rate their songs, because its time consuming, and also because users have afraid of inducing the program to not play or not include on a generated playlist, a particular song. This fact is interesting and calls into question some theories that use the feedback supplied by users to adapt the system.

The last part of our questionnaire tried to identify some users needs and what functionalities they would like to have in a new application for visualization and browsing of music collections. Users put the three suggested functionalities by the following order: efficient music search; interactive browsing; and automatic playlist creation using a set of criteria. Related to music search (in a multiple choice list), most users (54%) would like to search music using tags, 48% using mood and 46% by song similarity (figure 2). Concerning the automatic playlist generation, more than 51% of the participants would like to use genre as the main criterion. We believe that this is due to the small diversity of their libraries, allowing the application to cover most of their tracks. Additionally, they also would like to create playlist by providing tags (37%) and by selecting a mood (37%). Results provided by the online survey were later supported by the contextual inquiry, where we asked users to perform some tasks and to answer the same questionnaire, complemented with an informal interview.

Design Implications

From the previous results we identified three main design implications:

- **Browsing** The exploration of the music library should start by the selection of the Artist, then the Album, and finally the Genre;
- **Playlists** Users like to have control over the generation of playlist to make sure that some desired songs are on the list. To that end, we would like to provide some filtering

mechanisms to help users select the wanted tracks. This filtering can be based on tags and/or on music content;

- **Searching** To satisfy users’ requirements, we should provide at least three types of searching: i) using tags (those included in MP3 files, or added manually); ii) by selecting the mood (e.g. calm, energetic, etc.); and, finally, iii) by music similarity.

SOLUTION OVERVIEW

The solution we propose takes the form of a conventional desktop application, which meets the objectives and requirements set out in the introduction. The requirements identified can be divided into three types: visualization, retrieval, and extraction. This separation dictated the architectural decisions we took.

Based on what was said in previous paragraph, we decided to structure the system into three main components: a visualization component, which provides different views of the data model and several application controls (playlist, audio player, etc.); the information extraction component, responsible for both the extraction of semantic information (tags) and content-based audio features; and the retrieval component, that allows us to retrieve the information extracted by the extraction component. In the architecture defined, all components comply with an object-oriented model, and the Model-View-Controller (MVC) architectural pattern.

**Extraction Component**

The combination of metadata and content-based features is good for the construction of visualization systems for musical information, and our solution incorporates these two aspects. So, as said, the extraction component has two purposes: metadata and content-based extraction.

The metadata extraction component is responsible for extracting all the textual information associated with tracks, such as title, track length, album name, and artist, etc. This information, despite the potential problems of inconsistencies between them, allows us to develop a conventional text search engine, and an information display about music. It also helps in shaping the views over the music collection according to users’ preferences (Artist, Album, and Genre).

Because the focus of our solution is the end user of audio systems, the content-based feature extraction is not accomplished to individual features, but instead, to detect the similarity between songs, albums and artists. This component is responsible for determining for each track a descriptor that represents the information associated with it (similarity descriptor). We also contribute to similarity detection with an additional solution for album and artist similarity: the descriptors for these elements are determined from the average of music and albums descriptors.

This component is also responsible for the extraction of audio snippets, from the songs, albums, and artists. The snippet extraction is calculated using a heuristic algorithm, that divides the songs in their conceptual structure (intro; verse; chorus; verse; chorus, etc) and tries to determine where the chorus is. Preliminary results with five users showed that this algorithm works, having a 80% success rate within chosen 10 tracks.

**Retrieval component**

The Retrieval component is the element responsible for the recovery and retrieval of all the information extracted in the extraction component. This retrieval is accomplished in two ways: tags and content-based audio information.

The main goal behind semantic information retrieval (tags) is to guide the drawing of the visualization techniques. Additionally, tags are used to provide additional information to user about the songs, albums, and artists they are watching. To obtain information for each song there is an object (wrapper) that indexes the structure that holds the information (database), encapsulating how the data is actually stored.

Information retrieval of music content is more complex than the retrieval of semantic information. In this case, the similarity information, the descriptors processed in the extraction component are stored in a multi-dimensional structure that allows our retriever to perform \textit{knn queries} to obtain similar objects [4, 12]. The structure used is the NBTree presented in [3].

**Information visualization component**

This component is responsible for managing the interface and system interaction with the user. There are three key elements in this component: the two views (information visualization techniques) over the collection and the dynamic filter component. The other modules of this component are: the audio player, the playlist manager, and the filesystem explorer. This component holds the main contribution of this work: the interactive and dynamic filtering mechanism. Like the main treemap view (similarity ordered treemap), this module explores the treemap concept applying it to the visual representation of the filters.
In figure 3, we can view the internal architecture of this component. This image shows the main modules of this component and their interaction. Here all the modules are designed based on the MVC architectural pattern. This component is built from all the information extracted in the extraction component and only accessible through the retrieval component. In the next section, we describe the most important element of this component: the filtering module.

INTERACTIVE FILTERING MECHANISM
As stated before, this module is the most important contribution of our work. It is responsible for allowing the user to interactively filter your collection and explore it, progressively restricting the view over the collection. This element has seven basic filters, including: text, musical similarity (song, album and artist), duration, year, genre, beat, and mood. The choice of these filters is a direct consequence of the findings and implications extracted from the user study execute before the architecture design.

The interactive filtering is the solution designed to filter the collections examined by MuVis. Its general function, as mentioned in previous paragraph is to restrict the view over the collection, reducing the number of songs that the user sees at any given time, just to see the ones that meet the requirements (filters) specified.

We call this type of filtering dynamic and interactive filtering because the user controls at any time the applied filters (the order is irrelevant to our solution) and can add filters and remove others, seeing the changes in views almost instantly; users control the number of filtered tracks and the number of tracks that will be able to filter by selecting a filter. Using this feature, users can interactively explore their music libraries, applying a set of filters, but always controlling the state of the library and with full knowledge of their actions.

This module was designed by combining the architectural pattern cited above (MVC) with another design pattern: the Decorator. This pattern allows us to add behaviour to an object at run time without using the object extension mechanisms, but instead, using the composition mechanisms. The choice of this pattern to design the internal structure of this module was strongly influenced by the criteria of flexibility and extensibility we imprinted in our solution, and for what was described in previous paragraphs. These objectives are achieved through this pattern through the composition engine, which allows our solution to add and remove filters dynamically.

This module interacts directly with the component of information retrieval, and the retrieved information is used to describe the filters. The importance of this module in our solution is vital, since it enables not only to filter music collections, but also to explore and navigate them.

The internal representation of the filter element is based on a set of rules, restrictions that constrain the data to be displayed to user. This set of rules are applied directly on the raw data (directly in a relational database) allowing a fast filtering, and not requiring a set of media objects in memory to support browsing and filtering.

Each filter in this architecture acts as a single element of it. Thus, each filter has its own interface using the same paradigm of the principal display in ordered Treemap (View layer), an associated controller (Controller layer), and an element for creating the filtering rules (Model). Only the similarity and textual filters have different interfaces, but the remaining logic keeps applying.

The biggest challenge in developing this solution was the extensibility of the application on filters, to allow the users to control the part of the collections they see. In a more scientific point view, the most challenging part of this module was the combination of filters and the rules creation process, due partly to the implementation of the Decorator pattern. This mechanism, together with the View Manager, constitutes the biggest challenge of this work.

PROTOTYPE
Once defined, the architecture for our solution, we were faced with the decision on the development platform suitable for the prototype application. We decided to take the following main criteria: the portability of the development platform, the existence of toolkits for content-based audio features extraction, and the platform support for the creation of the main visualization technique (Semantic Ordered Treemap). After some tests and further investigation, we chose to develop our prototype with the Java platform. The chosen platform showed good portability (is currently supported within Linux, Windows and MacOSX Operating Systems), a diverse set of toolkits for audio feature extraction, and several libraries for the creation of TreeMaps with relative ease and extensibility.

In figure 5 is displayed the MuVis prototype interface with the treemap view active, and the visual representation of the several modules that compose our solution. On the left, we have the filesystem explorer and by its side the filtering panel. In the center we have the main view (in figure treemap is displayed, but a list view is also available), where
we can not only explore the collection, but also understand
the information that we are viewing. On the right, we have
the playlist manager, where the songs in the playlist are dis-
played and where users can perform playlist management
(remove songs, load or save playlists, etc.). Below is the au-
dio player, with a panel where textual information about the
song currently playing is displayed.

While users browse and apply filters, the view over the col-
lection adapts to only display the songs (artists) that match
the filters (see figure 6). In the treemap view, by double-
clicking in an artist, users can zoom in to explore more infor-
mation about the artist: its albums and tracks (see figure 7).
Seeing this detailed information, users can learn more about
that artist, and can even apply all the operations available
through all the application: preview tracks, albums, artists;
apply new filters, etc.

**EXPERIMENTAL EVALUATION**

After the development of the MuVis prototype, we conducted
an experimental evaluation. The evaluation described in this
section seeks to measure the achievement of the objectives
described in the introduction. This evaluation consisted of a
set of tests. Each test consisted of some typical tasks and a
survey.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Find the song Wonderful Tonight from Eric Clapton and ear it</td>
<td>Find a specific song</td>
</tr>
<tr>
<td>T2</td>
<td>Create a playlist with 12 songs, from the 90’s, Pop genre, and with duration between 3-5 minutes</td>
<td>Implement a set of filters to the collection and create a playlist</td>
</tr>
<tr>
<td>T3</td>
<td>Register the artist name with more songs and the artist name with more albums</td>
<td>Determine the artists with more songs and more albums</td>
</tr>
<tr>
<td>T4</td>
<td>Find similar albums to the first album of the artist with more albums</td>
<td>Perform filtering using the similarity between albums</td>
</tr>
</tbody>
</table>

We selected 10 participants representing the target user pro-
file for our system. This type of users usually listens to music
on their personal computer, and their libraries have a large
size. In a technological level, we sought people who had
knowledge in using software to listen to music.

The experimental evaluation consisted of four phases taking
approximately 35/40 minutes. First, the observer began by
explaining the context and objectives of our work, describ-
ing the steps and expected results to users. At this point we
presented three information items: the tasks and two demon-
stration videos, one for each application.

The next two steps in the experimental evaluation consisted
of performing the presented tasks in both applications (table
1). The tasks covered the main components of our solution,
and a progressively difficulty.

Each step was preceded by a period of training and adap-
tation to the program under evaluation. Users were divided
into two groups. One group used MuVis first and then Win-
dows Media Player (MuVis / WMP), while the other group
used the applications in reverse order (WMP / MuVis). We
have taken this methodology to obtain more reliable results
and also study whether there were differences in results. The
last step consisted of a inquiry and an informal interview.
Users completed a survey that sought to capture information
on user profile and experience. After the survey a short in-
terview was also conducted, followed by an open dialogue.

**Results**

The main results of the experimental evaluation reveal the
success and achievement of our objectives.

Observing the graph in figure 8, we can notice that there are
relevant differences in the execution time of the various tasks
in the two systems. While the difference between the execu-
tion time of the first task in both applications is not signifi-
cant (the task was indeed simple), concerning to other tasks,
this no longer happens. Users executed the required tasks
in MuVis much faster than in WMP (80% faster). In fact,
on the second task, the difference between applications (in both groups) is about three minutes. The main reasons that influence these results are different, but highlight the main advantages of our solution at the application benchmarking.

These results confirm what was also said by the users in the satisfaction surveys (see the section below), where the combination of filters and understanding of the library were not easy to achieve nor to understand in the Windows Media Player, unlike MuVis.

These results show also that the order in the assessment of the systems does not influence the performance of users in the tasks execution. The only exception corresponds to the search for similarity (Task 4), which highlights a clear difference between the WMP / MuVis and MuVis / WMP groups. In this case, users who have not used yet the search for similarity MuVis, were not conditioned and influenced in their concept of similarity. However, the observation of these results showed that users find it difficult to define the concept of similarity.

In the graph of figure 9 we can see the total execution time of the requested tasks. By observing this graph we confirmed that the tasks were executed much faster in MuVis than in WMP. On average, MuVis users were able to complete the requested tasks in two minutes, instead of the eight minutes in WMP. We also observed that users in WMP / MuVis group took more time performing the tasks in two applications: not only because they had difficult in understanding the organization of the collection, but also because they were confusing about defining similarity. The users frustration was clearly evident when doing the tasks.

Regarding the number of required steps for the tasks, this result clearly confirms the efficiency of our application over the comparative system. While in our application the average number of necessary steps to do the four tasks was 24,
As for the views used to perform the tasks, the results were encouraging and confirm that the treemap representation is appropriate and an effective way to explore music libraries (this is also supported by the observations made in the tests).

In both groups, the List view was only used in the first task, while the Treeview view was the most used in other tasks. This confirms the usefulness of this view to understand the global organization of the collection. On the other hand, in WMP, to perform the required tasks, users needed to use at least three different views (Track, Artist and Album), which are difficult to visualize, since it is not possible to simultaneously combine the information from the different views.

For the measure mark of the number of filtering operations required to complete the tasks, the results shown in figure 10 prove the assumptions of effectiveness required. To perform the same tasks, the number of filtering operations required in WMP is three times higher than the value required in MuVis. These values are explained by the potential of the filtering criteria combination provided by MuVis.

On our solution, using only one view, you can combine multiple filters (with a click for each filter) and see the results straightaway, unlike WMP, where the combination of filters is tricky, difficult to understand and with several restrictions. This is again justified by the execution time of all tasks, earlier described in this section. By observing users performing the tasks in WMP, we noted their frustration and difficulty combining filters to perform simple searches, unlike MuVis, where the operations were done in a simple, “intuitive” (referred by users in the questionnaires) and with total control way.

The functionality of audio preview was used by all users intensively, when they wanted to hear a song that they didn’t know, or as an aid to exploration and browsing. They were delighted at the effectiveness of this mechanism, where the chorus of the vast majority of the songs was played.

In short, users generally enjoyed MuVis. They were amazed at the simplicity, speed and effectiveness of our application compared to a known product as Windows Media Player. Users highlighted as innovative the filtering mechanism, the library visualization and browsing (treemaps), and previews of songs.

CONCLUSIONS

The obtained results during the experimental evaluation were presented in previous section and allow us to draw our conclusions. We noted that the various components of our solution were properly used and promoted by users to explore music collections. MuVis solution enables a faster exploration and interactive browsing for music libraries, and allows users to actually know them better. The survey and interview results corroborate this fact, where the participants stood out as advantageous to have an overview of the collection to know what with what they can count on.

Our contribution is stated as a new approach to interactive and dynamic exploration of music collections, which combines all the components and mechanisms described early in this article. Based on the results described in the preceding paragraph we conclude that our approach actually contributes to an increase in the ease, speed and flexibility in the exploitation of music collections. We conclude that the paradigm of similarity ordered treemaps, actually contributes to higher understanding of the music libraries, and thus allow finding songs and performing searches that would not be easy (or even possible) to perform with current systems.

The results from the experimental evaluation showed that a simple and effective combination of a set of filtering criteria (dynamic filtering) provided by a few interactions fully satisfies users. In addition to the speed and efficiency, the visual representation of the filters using the paradigm of treemaps also allowed the users to know their music libraries: this mechanism allows not only filtering but also helps in browsing the collection.

The architecture developed is also an important contribution. The solution was designed to be flexible and extensible at various points, through the use of several design patterns. This flexibility occurs within all of the three main components of the solution. Thus, the potentials for further work, both technical improvements, and implementation of new solutions can be driven by gains in the development and rapid implementation of these techniques.

No less important than the contributions cited in the preceding paragraphs, are the results presented in user study section. These results represent the basis of all the work developed, since the conceptual architecture to the functional prototype, guiding the final solution to be in accordance with the users requirements. The design implications stay registered as a basis for any future work that needs understanding of users needs to listen to music.
Based on the experimental evaluation performed and the constraints referred to throughout this work, we have identified some aspects that encourage the development of future work on this subject. We identify three main paths for future work development: the usability of the application, improvements in the mechanisms of extraction and information retrieval, and enhancements to the information visualization. However, because our focus was about visualization, the most likely path to follow is the about the information visualization. In this area, our future works might pass through the development of new visualization techniques or through the re-combination of some techniques (e.g.: Collect Tables [10]) in order to overcome the problems of the existent ones. This would involve refining all the steps described in this document, as new solutions can bring new benefits, but also new problems to.

ACKNOWLEDGMENTS
I thank all who contributed in the preparation of this work with their true support. To all who participated in this project a big thank.

REFERENCES