VISUALIZING AND EXPLORING PERSONAL MUSIC LIBRARIES

Marc Torrens, Patrick Hertzog
Artificial Intelligence Lab.
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerand
{Marc.Torrens, Patrick.Hertzog}@epfl.ch

Josep-Lluís Arcos
Artificial Intelligence Research Institute
Spanish Scientific Research Council
08193 Bellaterra, Catalonia, Spain
arcos@iiia.csic.es

ABSTRACT

Nowadays, music fans are beginning to massively use mobile digital music players and dedicated software to organize and play large collections of music. In this context, users deal with huge music libraries containing thousands of tracks. Such a huge volume of music easily overwhelms users when selecting the music to listen or when organizing their collections.

Music player software with visualizations based on textual lists and organizing features such as smart playlists are not really enough for helping users to efficiently manage their libraries. Thus, we propose new graphical visualizations and their associated features to allow users to better organize their personal music libraries and therefore also to ease selection later on.

1. INTRODUCTION

New technologies combining portable digital music players with dedicated software (such as iPod [1] with iTunes [2]), together with new music distribution channels through Internet are quickly changing the way people organize and play music. Thus, a new community of digital music users is emerging. These users deal with music differently compared to the traditional way. Instead of dealing with albums or CDs, they basically face their music at the track level by :

- acquiring track by track, and
- creating and playing personalized playlists.

In such contexts, users have to deal with huge libraries of music in their computers and mobile players. Music libraries can easily contain thousands of tracks (corresponding to hundreds of CDs). Such a huge volume clearly overwhelms users when choosing the music to listen at a certain moment. Therefore, this situation poses several IT challenges regarding how to offer adequate tools to users in order to support them organizing their collection and in their decision making process of selecting, and playing music.

[1] `http://www.apple.com/iPod`.
[2] `http://www.apple.com/iTunes`.

1.1. Basic Notions

A digital music library usually refers to a set of tracks in any electronic format [3] . However, in this paper we are only interested in track descriptions, *i.e.,* in the semantic attributes associated to tracks. In the following, we describe some basic notions that will be used throughout the paper.

A music *library* is a set of descriptions of every track acquired by a user. A track description contains attributes such as artist, composer, year, album, genre, and so on. Usually, a music library is stored in a XML file.

Most of the music player software offers the possibility to create sequences of tracks allowing users to group tracks and therefore facilitate the task of selecting which track to play next. A *playlist* is a subset of a library and defines an ordered sequence of tracks to be played. Playlists are usually created by selecting tracks one by one. On the other hand, a *smart playlist* is a playlist which follows a set of logical filtering criteria. Smart playlists are useful for grouping tracks under a certain logic. For instance, a user may specify a playlist with all the 70's jazz tracks which were played in the last 6 months.

Playcounts are automatically generated by music player software in order to trace how many times a track in a library has been played. It is also common to consider users' ratings for specific tracks. Such rating is used to express a certain degree of preference of a track over the others in the library.

1.2. Standard Tools to Visualize and Manage Libraries

The most basic way of visualizing music libraries is using textual lists where each item of the list shows the attributes of one track. For example, *iTunes* [4] proposes ordered lists as a way of visualizing, managing and browsing music.

[3] such as MP3, AAC, etc.
[4] Throughout this paper, we consider *iTunes* as the reference and leader of music player software in the market.

As shown in Figure 1, track lists can be ordered alphanumerically by any of the attributes, *e.g.*, by genre, artist, composer, album, year and so forth.

1.2.1. Searching by keywords

A simple mechanism for searching a specific track in a music library is through a keyword-based search functionality, as offered by *iTunes*. Users can enter a keyword and then specify if the search has to be applied to all attributes or just to one of the predetermined ones (namely artists, albums, composers or track names).

1.2.2. Searching by filters

In *iTunes* users can also search tracks by selecting three kind of filters: genres, artists and albums. The three filters work with multiple selection and there is no order imposed by the software. In a results window, the filtered tracks are displayed. Users can then order the tracks alphanumerically (ascendent or descendent) by any of the attributes.

1.2.3. Standard playlists and smart playlists

In order to organize and manage music libraries, *iTunes* offers the possibility of creating playlists in two different ways: 1) to add tracks in a standard playlist by manually selecting them one by one from the library, or 2) to define a set of filtering criteria from which the smart playlist is created. These filtering criteria refer to one of the track attributes and are of the type *contains*, *is*, *starts*, *ends*, *is before*, etc. depending on the selected attribute. Other parameters such as the size of the playlist can also be specified, for instance, by the number of tracks, the duration of all tracks, etc. If such parameters are specified, then a selection argument can be chosen such as *random*, *most played*, *highest rating*, and similar others. Finally, *iTunes* allows users to decide if a smart playlist should be lively updated or not. A smart playlist with live updating is a playlist which is updated according to new tracks added to its associated music library.

1.3. Related Work

To our knowledge, there is not so much work done on visualizing and exploring personal music libraries based on the semantic attributes of tracks. However, there is a relevant research community working on how to visualize and organize music based on signal processing techniques. In that sense, *Islands of Music* [8, 10] organizes music libraries without requiring genre or other attribute classification because it uses psycho-acoustic models, and then tracks are visualized using a metaphor of geographical maps where islands resemble genres of music styles. A different approach using a heuristic version of Multidimensional Scaling (MDS) named *FastMap* is described in [1]. With a similar goal to *FastMap*, *Sonic Browser* [6] uses sonic spatialization for navigating into audio libraries and the Marsyas3D tool [12] proposes techniques based on principal component analysis for browsing audio libraries.

Another related work is enlighten in [2] but it is more focussed on how to identify any potential misfits between the designers' views of a product or system, embodied in the device itself, and those of its users. Our work has another point of view since we do not consider how to build the whole system but just how to visualize and manage the existing music libraries of users.

On the other hand, Beth Logan presented in [3] an approach to form playlists from a given seed song. Their technique is based on their own audio content similarity measure introduced in [4]. Paws and Eggen [9] propose to generate automatically playlists with an inductive learning algorithm considering different context-of-use of music consumers.

All these previous works are based on audio information and use signal processing techniques. Another related work is the *Variations2* project [5, 7] at the Indiana University. They exploit music bibliographic data for providing visualization methods in order to assist music students and faculty members.

The work presented in this paper is radically different focussing on the attributes associated to the tracks.

1.4. The challenge

Music player software with visualizations based on textual lists and organizing features such as smart playlists are not enough for helping users to efficiently manage their libraries which may easily contain thousands of tracks. Thus, in order to avoid the current situation where users are clearly overwhelmed with the problem of selecting tracks, we propose new visualizations and their associated managing features. In the following, we describe our proposals and then we compare and evaluate them.

We have basically explored three [5] different visualizations which allow users to have a better overview of the contents of their music libraries and therefore to ease its organization. On the other hand, the two first visualizations are shown to be very useful helping users to build playlists graphically instead of having to express filtering criteria which may may be confusing to users.

2. VISUALIZATION TECHNIQUES

This section presents three different ways of graphically visualizing music libraries considering five criteria (thus, five dimensions) which are genre, artist, year and a quantitative criterion to be chosen by the user such as playcount, rating, added or last played date. The goal of these visualizations is two-fold: a) to give an overview of the contents of a music library, and b) to visualize playlists and give some support to manage and organize them. Depending

[5] As we will see, the two first visualizations, using discs and rectangles, can be considered as variants of the same basic concept.

on the visualization model users get different advantages since they have a different geometric expressiveness. All the explored techniques give a topologic overview of a music library regarding its tracks.

2.1. Disc Visualization

This visualization, called disc visualization, is based on well-known visualization charts in form of discs. Users are used to manage such kind of visualizations which give good percentage and proportional overviews. However, this visualization is different than standard pie charts as we will see in the following sections.

2.1.1. Description

As shown in Figure 2, the disc is divided in different sectors that represent each of the genres of the library [6] . The size of a sector is proportional to the number of tracks of the associated genre with respect to the whole library. Therefore, the size of a sector is directly proportional to the importance of the corresponding genre within the library. At the same time, sectors are split in sub-sectors representing the artists of the associated genre. Again, the size of sub-sectors is proportional to the number of tracks of the artist. The radius of the disc, from the center to the perimeter, can be seen as the time axis: the center represents the year of the oldest track of the library and over the perimeter the most recent tracks are positioned. Tracks are then depicted as points over the disc according to their attributes, *i.e.*, genre, artist, year. Tracks belonging to the same album are positioned contiguously, thus it has the effect of producing like arcs of points representing albums. The order in which the albums are depicted is alphanumeric, and the order for the tracks of the same album is the original order in the album. The quantitative attribute to be chosen by the user (for instance playcount, rating, last played date, added date, etc) is depicted according to different color tonalities. Colors are used to express the exact value for one track in its associated point. The mean value of all the tracks for one genre is also used to color the corresponding sector.

Figure 3 illustrates how playlists and smart playlists can be shown using the disc visualization. Tracks of playlists without any grouping logic can be depicted by using geometric forms different than regular points which are used in general for the rest of the songs. For instance, in Figure 3, the playlist called “Jogging playlist” (number 4) is displayed by using bigger points in red. The other example in the figure for playlists which do not follow any geometric logic is the “25 last played” (number 3) whose tracks are represented as little red crosses. The other playlists (numbers 1, 2, and 5) are shown as red regions since they follow a regular geometric form. The rest of the playlists (numbers 6 and 7) are not highlighted since the user has not activated their corresponding checkboxes.

[6] The figures illustrating the disc and rectangle approaches were generated with a real music library of about 2.500 tracks.

In such a visualization, the track currently being played could be highlighted and a path grouping the tracks to be played next could also be displayed. In this way, the user will get an idea of what regions of his library are going to be used in the current music sequence.

2.1.2. Interaction Principles

In this section, we describe how users interact with the disc visualization. Basically, the following principles have been identified:

Navigation. The attributes of any track of the library can be visualized in textual form by just positioning the cursor over its point. For example, in Figure 2, the cursor is over the track *I don’t want a lover* by *Texas*, so the song attributes are displayed in the bottom left corner. Also, when moving the cursor over the disc, the artist of the corresponding sub-sector is highlighted as shown in Figure 2 for the *Texas* group. In a similar way, the year is indicated with a circle, as illustrated in the example of Figure 2 with the circle of the year 1989.

Zoom. When a user is interested in getting a more detailed view of his library, he can zoom over any sector of the disc. This zoom will then generate a disc with the same visualization and interaction principles but applied to just the genre of the selected sector. Therefore, in this first zoom level, sectors representing genres become sectors representing artists with sub-sectors representing albums. All the other dimensions and general principles remain the same. Finally, the latest zoom level when selecting a sector representing an artist would produce a disc where sectors are the albums of the selected artist, without any sub-sector. Thus, in this latest zoom level, users obtain a graphical representation of the tracks for a given artist.

Playlist management. As explained in Section 2.1.1 and shown in Figure 3, the disc visualization can be nicely used to graphically display playlists and smart playlists. Moreover, this visualization can be used to edit or create new playlists with useful graphical help. The mechanism is based on considering playlists as sets and then being able to construct set operations to form new playlists. Multiple playlists can be selected at the same time and then apply operations such as union, intersection, difference, and so on. Since playlists are graphically visualized as sets, it is convenient and useful to apply set operations over them. The resulting playlists are also graphically displayed.

When creating (or editing) playlists with tools like the ones provided by *iTunes* (either directly selecting songs, or by constructing a set of logic rules for smart playlists), the disc visualization is useful for showing the playlist being created step by step. So, at any moment of the creation (or edition) of a playlist, the user can immediately see how the new playlist changes, its approximate size and its topology. Such procedures help users to have a better idea of which zones of the library are overused or underused, or the zones implied in each playlist.

Standard search procedures. When using standard search procedures like the ones described in Section 1.2, the disc visualization can also be of a great help by highlighting the filtered songs dynamically. In the same way that *iTunes* dynamically changes the list of tracks in the results window, the visualization highlights the tracks graphically.

2.2. Rectangle Visualization

This visualization is similar to the disc visualization but using rectangles instead of discs. In the disc visualization, the time axis was represented along the radius of the disc, and in the rectangle visualization the time axis goes along the vertical axis. Similarly, for this visualization, the attribute genre goes along the horizontal axis. The result of this visualization is shown in Figure 4.

Even if both visualizations have similar features, they may give different user experiences with their advantages and downsides as described in Section 3.

2.2.1. Interaction Principles

The main principles described for the disc visualization apply to this visualization, however the zoom functionality may be differently applied.

Zoom. In this visualization, zooms can be done in the same way as for disc visualizations. When zooming over a genre (which is a sub-rectangle), the horizontal axis becomes the artist dimension. Similarly, when zooming over an artist, the horizontal axis becomes the album dimension.

Another way of applying zooms in the rectangle visualization is to just consider that all the tracks in the library are always shown, but the scale of the horizontal axis changes. Therefore, using this approach, the user explores the whole library just by using a scroll bar for panning over an specific zone. In this case, when zooming in, the horizontal axis still represents the genres, and the artists within each genre. With a second level of zoom, in addition to genres and artists, the axis also represents the albums for each artist. The horizontal axis and its scroll bar are accordingly adapted depending on the zoom level.

2.3. Tree-Map Visualization

This visualization is using Tree-Maps in a similar way as described in [11] but for visualizing music libraries.

Figure 5 shows three different levels of zoom for the same library. In this visualization [7], the size of rectangles are always proportional to the number of tracks in the attribute represented by the rectangle. At the same time, rectangles are recursively split in sub-rectangles showing other proportions. For example, in Figure 5(a) rectangles are recursively split three times: the whole library (the parent rectangle) is split into genres, each genre is split in its sub-genres, and finally each sub-genre is split in its artists. Figure 5(b) is showing the genre *Rock*, so there are rectangles representing *Rock* sub-genres like *Rock and roll*, *Alternative*, and so forth. At the same time, these rectangles are split into the artists of the associated sub-genre. Figure 5(c) illustrates the visualization of the *Rock and roll* sub-genre, so rectangles representing artists are shown without anymore splitting. One could make a further recursion and split each artist rectangle by her/his albums.

The color of each rectangle indicates a quantitative attribute to be chosen by the user, similarly as the previous visualizations, *e.g.*, playcount, last played date, ratings, and so on. However, in this visualization, since tracks are not depicted, only mean values are represented by different color tonalities.

The interaction mechanism for the Tree-Map visualization is very straightforward for zooming: the user selects a rectangle, and the parent rectangle shows then the selected attribute.

[7] For better showing the concept of Tree-Maps visualizations for music libraries, we assume that a sub-category of genre, called sub-genre, is available for each track.

3. COMPARISON

The visualizations using discs and rectangles basically offer the same functionalities, while the Tree-Map visualization is more likely to be used just for giving a better overview of the contents of music libraries. This is because the disc and the rectangle approaches are capable to show information at the track detail whereas it is unclear how to represent tracks using Tree-Maps. A comparison among the different presented visualizations and their features is summarized as follows:

- Visualizations based on discs and rectangles offer similar functionalities, but also different pros and cons due to their different geometric forms:
  - Discs give a better visual idea about the proportions of sectors and sub-sectors compared to rectangles and sub-rectangles.
  - Track points are differently distributed in discs and rectangles. For libraries with more recent tracks than old ones, the points are better placed in the disc visualization. On the other hand, libraries which are more homogeneous with respect to the year of their tracks are better suited for the rectangle approach.
  - The zooming feature is more useable with rectangles since the whole library space can be represented with the help of scroll bars. Zooming in the disc visualization implies to focus to a smaller portion of the library.
  - In the rectangle version, both coordinates are visible (genre/artist/albums and year) thus the positioning of tracks is easily understood by users. In the disc representation, the year coordinate goes along the radius of the disc so possibly more efforts could be required by users to quickly understand it.

- Tree-Map visualizations are more adequate to give an overview with respect to the number of tracks belonging to each attribute represented by the size of its rectangles.
- The Tree-Map approach is not very well-suited for displaying information about tracks or playlists.
- Discs and rectangles can be used to visualize, and more importantly to create and edit playlists. The Tree-Map representation does not offer this possibility because tracks are not shown.

All the approaches presented in this paper (and also textual lists) should be regarded as complementary by considering the above arguments. In this way, a complete music player software may allow users to choose among the different approaches. Also, it is feasible to automatically decide which approach has to be used depending on the topology of the library of the user and the action the user is considering, resulting in a really *smart* music organizer.

4. CONCLUSIONS

Textual list-based visualizations and organizing features such as smart playlists are not enough to really support users who deal with music collections of thousands of tracks. In order to assist music fans to better manage huge digital music libraries, we have proposed new visualizations and their associated features. However, the proposed approaches should be regarded as complementary to more conventional tools like textual lists.

We believe that advanced but yet simple visualizations are critical for supporting the process of exploring and therefore re-discovering personal music collections. Actually, it seems reasonable to believe that many times users may be interested in rediscovering their own music instead of thinking about enlarging their collections. Moreover, users may be interested in exploring their music collection to actually decide what to acquire or listen next. Currently, this rediscovering process can be tedious by using textual lists, while the presented new approaches facilitate such task.

4.1. User Studies

Stronger and final arguments for validating the suggested approaches should be given by rigorous user studies. These user studies will be developed with different type of users considering at least factors like technology matureness, age, educational background, and topology of their libraries (size, recency of tracks, homogeneity).

Acknowledgments

We would like to thank Michel Speiser for his great implementation of the ideas shown in this paper during a semester project at the EPFL. We also thank MusicStrands S.A. for providing XML parsers to read real *iTunes* libraries.

5. REFERENCES

[1] Pedro Cano, Martin Kaltenbrunner, Fabien Gouyon, and Eloi Batlle. On the Use of FastMap for Audio Information Retrieval and Browsing. In *International Conference on Music Information Retrieval (ISMIR 2002)*, Paris, France, October 2002.

[2] Iain W. Connell, Thomas R.G. Green, and Ann E. Blandford. Ontological Sketch Models: highlighting user-system misfits. In P. Palanque E. ONeill and P. Johnson, editors, *Proceedings of Human Computer Interaction (HCI)*, Bath, England, September 2003. London Springer.

[3] Beth Logan. Content-Based Playlist Generation: Exploratory Experiments. In *International Conference on Music Information Retrieval (ISMIR 2002)*, Paris, France, October 2002.

[4] Beth Logan and Ariel Salomon. A Music Similarity Function Based on Signal Analysis. In *IEEE International Conference on Multimedia and Expo (ICME)*, Tokyo, Japan, August 2001. IEEE Press.

[5] Indiana university digital music library project. http://dml.indiana.edu/.

[6] D.O. Maidin and M. Fernstrom. The Best of Two Worlds: Retrieving and Browsing. In *Conference of Digital Audio Effects*, 2000.

[7] Mark Notess and Natalia Minibayeva. Variations2: Toward Visual Interfaces for Digital Music Libraries. In *Second International Workshop on Visual Interfaces to Digital Libraries*, 2002.

[8] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Content-based Organization and Visualization of Music Archives. In *ACM Multimedia*, pages 570–579, Juan les Pins, France, December 2002.

[9] Steffen Pauws and Berry Eggen. PATS: Realization and User Evaluation of an Automatic Playlist Generator. In *International Conference on Music Information Retrieval (ISMIR 2002)*, Paris, France, October 2002.

[10] A. Rauber, E. Pampalk, and D. Merkl. The SOM-enhanced JukeBox: Organization and visualization of music collections based on perceptual models. *Journal of New Music Research*, 32(2):193–210, 2003.

[11] Ben Shneiderman. Tree Visualization with Tree-Maps: 2-d Space-Filling Approach. *ACM Transactions on Graphics*, 11(1):92–99, January 1992.

[12] G. Tzanetakis and P. Cook. Marsyas3D: A Prototype Audio Browser-Editor using a Large Scale Immersive Visual Audio Display. In *International Conference on Auditory Display*, 2001.

Figure 1. Managing libraries with *iTunes* based on textual lists.

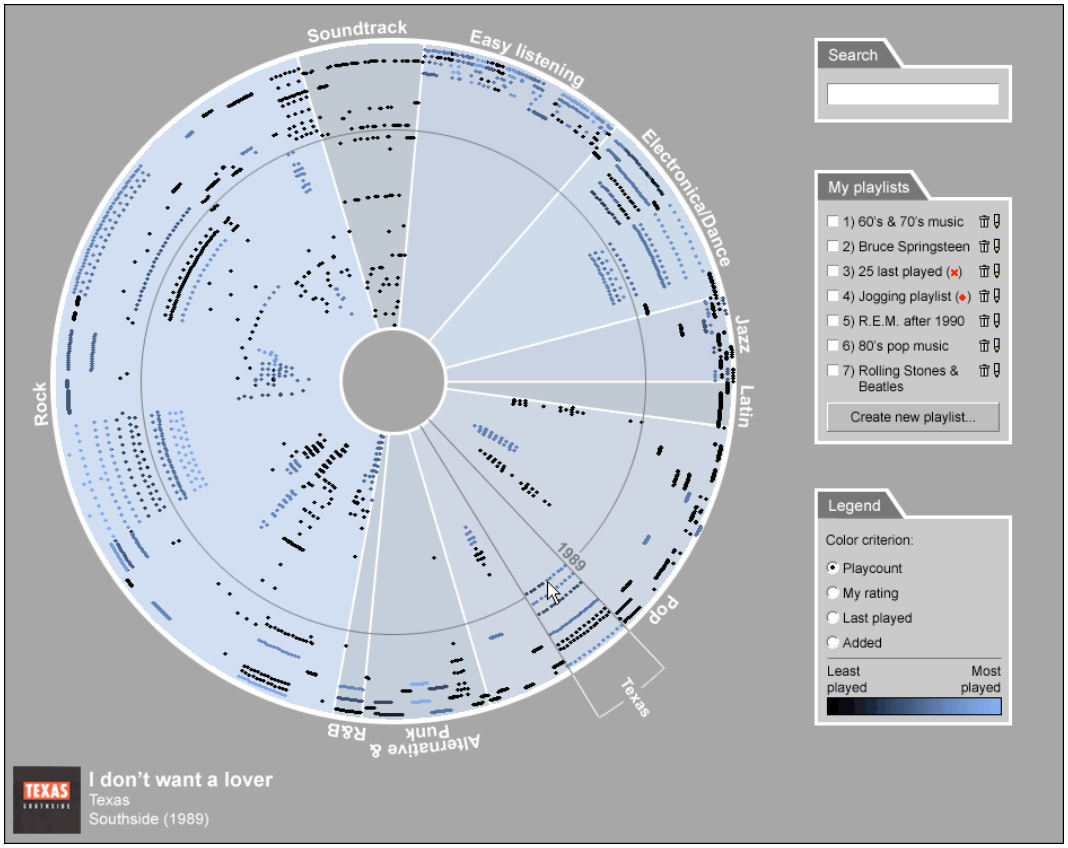


Figure 2. Visualizing music libraries by using discs.

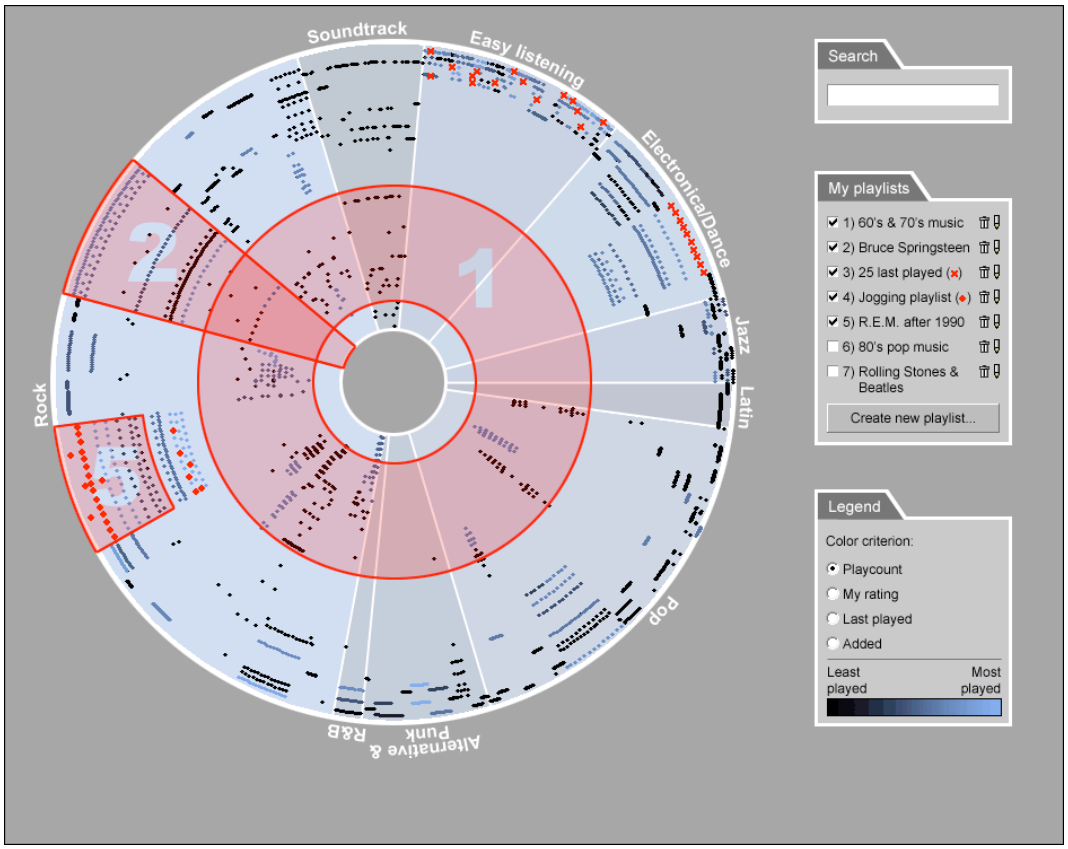


Figure 3. Visualizing playlists and smart playlists in music libraries by using discs.

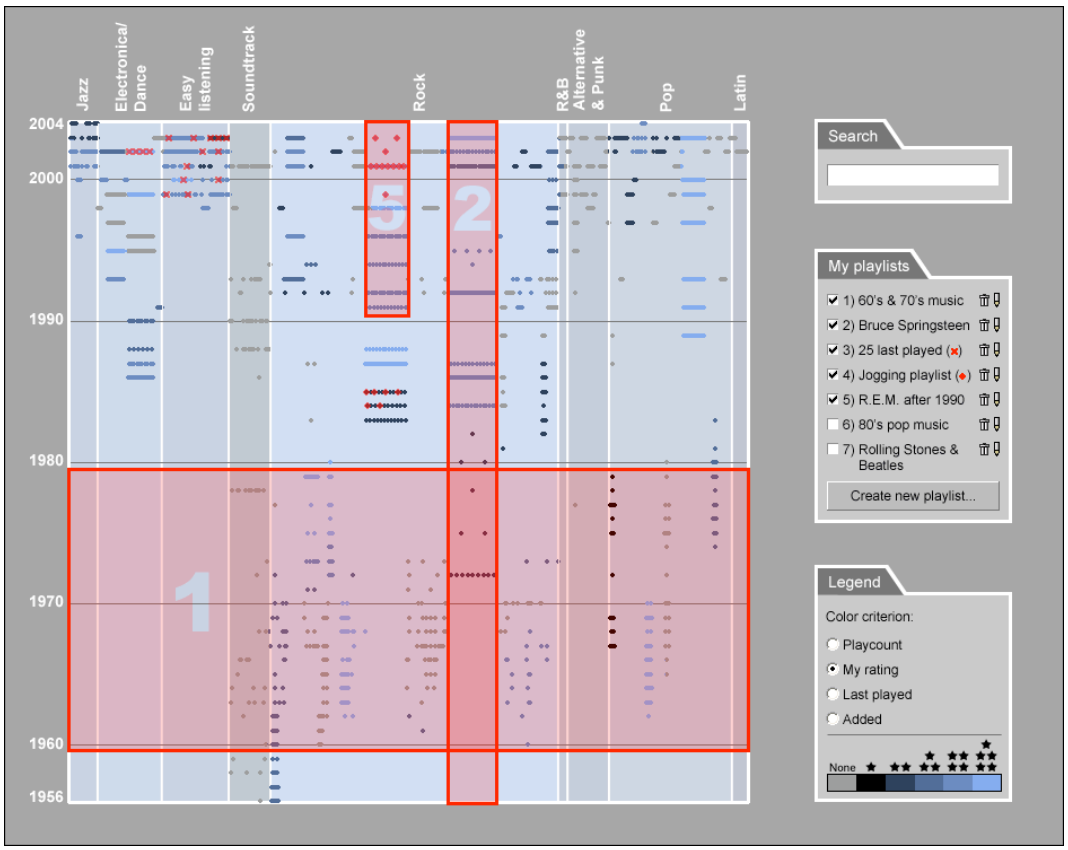
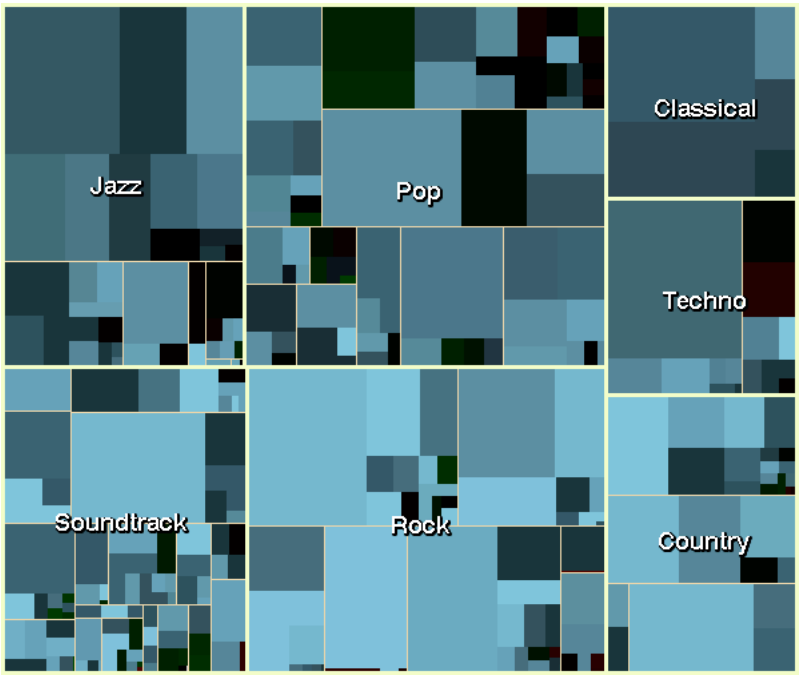
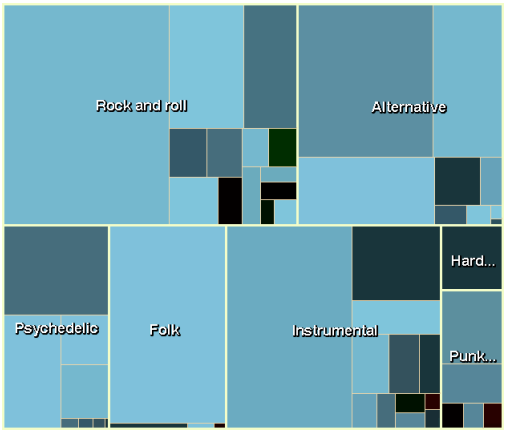


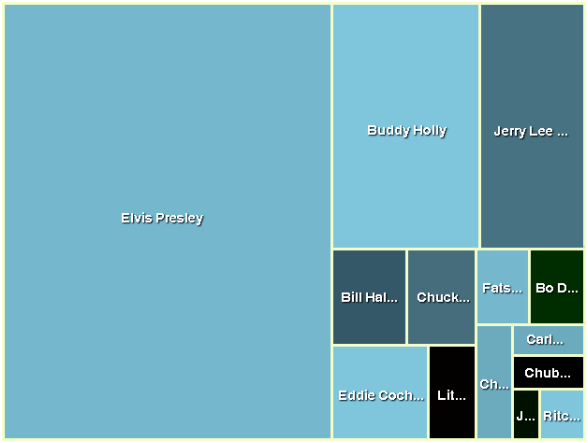
Figure 4. Visualizing music libraries by using rectangles.



(a) Tree-Map visualization for a whole music library.



(b) Tree-Map visualization for the *Rock* genre.



(c) Tree-Map visualization for the *Rock and roll* sub-genre

Figure 5. Tree-Map visualizations applied to music libraries. Different zooms levels are shown for the same library.