

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
Department "Institut für Informatik"
Lehr- und Forschungseinheit Medieninformatik
Prof. Dr. Heinrich Hußmann

Diplomarbeit

Visualizing Music Listening Histories

Frederik Seiffert

ego@frederikseiffert.de

Bearbeitungszeitraum: 1.10.2009 bis 31.3.2010
Betreuer: Dipl.-Medieninf. Dominikus Baur
Verantw. Hochschullehrer: Prof. Dr. Andreas Butz

Kurzzusammenfassung

Privates Musikhören ist in den letzten Jahren mit der Ausbreitung von digitalen Musikabspielgeräten scheinbar allgegenwärtig geworden. Dienste wie Last.fm ermöglichen es uns den Titel jedes gehörten Liedes zu erfassen, wodurch große Mengen an persönlichen Daten entstehen. Bis jetzt gibt es jedoch keine Mittel um die vorhandenen Daten umfassend zu analysieren. In dieser Diplomarbeit wird LastHistory vorgestellt, eine interaktive Visualisierung für den Verlauf der gehörten Musik, zusammen mit Kontextinformationen von persönlichen Fotos und Kalendereinträgen. Es lässt sich für zwei Aufgaben verwenden: Der „Analysis Modus“ ermöglicht Einsichten in beliebige Musikverläufe in drei grundlegenden Dimensionen: Zeit, Musiktitel, und Genre. Der „Personal Modus“ zielt darauf ab, die eigene Geschichte zugänglich zu machen, und unterstützt Reminiszieren durch den persönlichen Kontext in der Visualisierung.

Die Diplomarbeit zeigt zuerst verwandte Arbeiten zu Zeitleisten, Musikverläufen, und Interfaces zum Medien-Browsing, bevor sie das konzeptuelle Design, den Einsatz, und die Implementierung von LastHistory erläutert. Um die Arbeit zu evaluieren wurden zwei Benutzerstudien mit sehr positiven Ergebnissen durchgeführt, die am Ende der Arbeit vorgestellt werden.

Abstract

In the last years with the advent of digital music players, private music consumption has become seemingly ubiquitous. Services like Last.fm give us the chance to record every track we listen to, resulting in large sets of personal data. However, so far there are no tools for comprehensively analyzing the data at hand. This thesis presents LastHistory, an interactive visualization for displaying music listening histories, along with contextual information from personal photos and calendar entries. It is suitable for two tasks: The Analysis Mode allows to generate insight for arbitrary listening histories in three basic dimensions: time, tracks, and genre. The Personal Mode aims at making one's own history accessible, and encourages reminiscing through the personal context included in the visualization.

The thesis first shows related work on timelines, music listening histories, and media browsing interfaces, before outlining the conceptual design, operation, and implementation of LastHistory. Two user studies were conducted in order to evaluate the work, resulting in very positive results, which are presented at the end of the thesis.

Aufgabenstellung

Titel: Visualizing Music Listening Histories

Hintergrund: Bisher gehörte Musik ist eine reichhaltige Informationsquelle, die ohne Zutun des Nutzers automatisch mitgeloggt werden kann. Da diese Daten jedoch sehr komplex und umfassend werden können ist es schwierig sie mit Listen zu begreifen. Bisherige Ansätze zur Visualisierung kratzen nur an der Oberfläche.

Ziel: Im Rahmen dieser Diplomarbeit sollen verschiedene Konzepte zur Visualisierung dieser Daten erarbeitet und teilweise implementiert werden. Dabei soll von Prinzipien der Informationsvisualisierung aber auch des Data Mining (zur Reduktion der Datenbasis) Gebrauch gemacht werden. Abschließend soll durch eine Nutzerstudie die Anwendbarkeit der gefundenen Visualisierung ermittelt werden.

”Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.“

München, den 18. März 2010

Frederik Seiffert

Contents

1. Introduction	1
2. Related Work	3
2.1. Timelines.....	3
2.2. Music Listening Histories.....	6
2.2.1. Visualization.....	6
2.2.2. Analysis.....	13
2.3. Media Browsing Interfaces and Concepts.....	14
3. Conceptual Design	19
3.1. Music Listening Histories	19
3.2. Design Principles	20
3.3. Visualization of the Music Listening History	20
3.4. Visualization of Personal Streams.....	22
3.5. Interactive Usage.....	23
3.5.1. Navigating and Scaling the Timeline.....	23
3.5.2. Accessing Information about History Entries.....	23
3.5.3. Highlighting Identical Tracks and Track Sequences.....	24
3.5.4. Track Playback	25
3.5.5. Searching.....	26
4. Application Operation	27
4.1. Analysis Mode	27
4.1.1. Insights in the Time Dimension	27
4.1.2. Insights in the Tracks Dimension	28
4.1.3. Insights in the Genre Dimension	29
4.2. Personal Mode.....	29
4.3. Examples of Music Listening Histories	30
5. Application Design and Implementation	35
5.1. General Design.....	35
5.2. Data Retrieval from Last.fm	36
5.3. Data Model and Data Storage	37
5.4. Genre Classification.....	38
5.5. Visualization.....	39
5.5.1. Listening History Stream.....	40
5.5.2. Photo Stream.....	44
5.5.3. Calendar Stream.....	47

5.6. iTunes Integration	47
6. Evaluation	49
6.1. User Study.....	49
6.1.1. Study Insights in the Time Dimension	50
6.1.2. Study Insights in the Tracks Dimension	50
6.1.3. Study Insights in the Genre Dimension	51
6.1.4. Using the Personal Mode.....	51
6.2. Online Survey	52
6.2.1. Application Usage Results.....	52
6.2.2. Evaluation Results	55
6.2.3. User Feedback	56
7. Conclusions and Future Work	59
8. References	61
9. Web References.....	63
10. Appendix A: Online Survey.....	65

1. Introduction

In the last years with the advent of digital music players, private music consumption has become seemingly ubiquitous: thanks to portable and ever-shrinking music devices, everywhere in public we now see people with headphones listening to music. At the same time, distribution of digital music over the Internet has become commonplace and has made an ever-increasing amount of music available to listeners in digital form. Now being part of the digital domain, music listening has become easily trackable. In this context, the practice of *life-logging* has recently been developed to result in an „undiscriminating collection of information concerning one’s life and behavior“ [1]. As shown by the *MyLifeBits* project [2], the original issue of recording and storing these large amounts of data can now be considered as solved if restricted to the digital world. Now the question of how to access and use this information arises, which constitutes the challenge taken in this work.

In the domain of music, services like Last.fm are able to log the personal consumption of music, resulting in a chronological collection of musical items listened to by a person: personal *music listening histories*. The process of logging this data is called „scrobbling“ by Last.fm and accomplished by simply installing a background application on one’s computer, which transmits information about each song played through both desktop-based and portable music players to a central server [24]. The resulting profiles can easily contain tens of thousands of entries, e.g. for people who constantly listen to music during work.

Being a rich data set that directly relates to a person’s life, listening histories can be used for two tasks: they can be analyzed for patterns in a person’s behavior, and they can be used by the creator of the listening history to revisit the past for reminiscing, and also for self-presentation and telling others stories of one’s life (see [3]). While the former follows from the data set being created as part of our daily routine, the latter can be explained by looking at music as a form of memory cue. As noted by van Dijck, music can work as a trigger, „bringing back waves of emotion, the specificity of a time, an event, a relationship, or evoking more general feelings“ [4].

The means by which this thesis aims at facilitating the two use cases of analyzing and reminiscing is information visualization, „the use of computer-supported, interactive, visual representations of abstract data to amplify cognition“ [5]. By visualizing music listening histories, the primary goal of this work is to enable their analysis to identify patterns in a person’s music listening behavior. As proposed by Pousman et al. [6], the visualization should be accessible to regular users and allow them to analyze their own data, which also fits with the more personal nature of the second use case.

This personal use case of reminiscing and storytelling represents the secondary goal of the visualization. In that regard it has to be noted that listening histories contain only songs and timestamps, and are void of any personal information to help make sense of discerned patterns in the context of a person’s life. Therefore, this use case suggests enhancing listening histories with additional information from a person’s past that allows integrating it into a personal narrative. As shown by Ringel et al. [7], photos and calendar entries can pose as personal landmarks for this task.

This thesis is structured as follows: Chapter 2 reviews related work on timelines, music listening histories, and media browsing interfaces. Chapter 3 first delineates the underlying data of music listening histories that can be analyzed through the visualization, before describing the design principles and the overall interface- and interaction design of the application. Chapter 4 outlines the two application modes – the Analysis Mode and the Personal Mode – that are optimized for the two use cases analysis and reminiscing respectively, and how they can be used for different tasks and to generate different kinds of insight. Chapter 5 explains the application design and some details about the implementation. Chapter 6 describes two user studies that were conducted in order to evaluate the work, and presents their very positive results. And finally in Chapter 7, a conclusion is drawn and some examples for future work are provided.

2. Related Work

This chapter describes related work on the visualization of timelines and music listening histories, as well as other media browsing interfaces.

2.1. Timelines

Temporal data is common for visualizations, and as such, various approaches to visualizing temporal data exist for different data sets and use cases.

LifeLines

LifeLines [8] provides a general-purpose visualization for personal timelines in medical or court records or other types of biographical data. Different aspects of the data are grouped into individual timelines, within each of which aspects of the data such as medical conditions or legal cases are displayed as horizontal lines. Color is used to indicate relationships between different events, and the thickness of the lines indicates their significance (see figure 2.1). Interactive tools like scaling and filtering can be used to focus on part of the data.

The tool was initially developed as a replacement for a textual form-based application tracking cases of delinquent youth behavior. According to the authors, the main benefits of LifeLines over the existing applications were that it reduces the chance of missing information and facilitates the spotting of anomalies and trends.

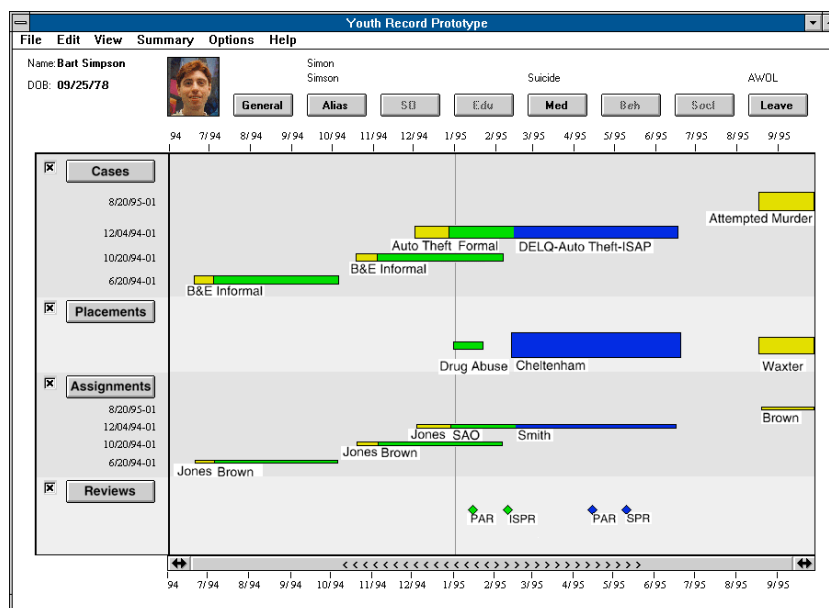


Figure 2.1. LifeLines screenshot

Time-Series on Spirals

Time-series data often appears in a cyclic manner: while time moves forward serially, days, weeks, months, and years recur periodically. Depending on the data set, other periods can occur like seasons (e.g. in temperature data) or projects (in project-based time tracking). Carlis and Konstan presented a visualization technique for displaying both serial and periodic attributes of a data set through a spiral data layout: while serial data is exposed along the spiral, periodical data is placed radially such that each period fills 360 degrees (one lap) of the spiral [9].

Figure 2.2 shows as an example the monthly food consumption of chimpanzees during a 9-year timeframe. Here, one lap of the spiral is equivalent to one year, while each spoke equals one month. The size of a blue blot corresponds the amount of food consumption during the month. Using this visualization, primatologists were able to recognize periodic patterns and anomalies within the patterns.

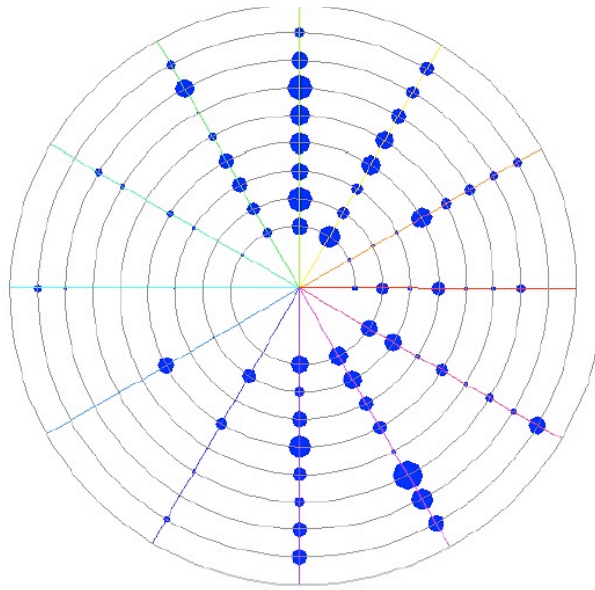


Figure 2.2. Monthly food consumption of chimpanzees visualized on a spiral

Cluster-based Visualization

Van Wijk and van Selow presented a method to identify patterns and trends simultaneously on multiple time scales (days, weeks, seasons) [10]. They first show a visualization of power-demand where the time scales are visualized explicitly: by considering time as two-dimensional as a function of hours and days, these scales are mapped on different axes, while the data is visualized using color (see figure 2.3). While patterns in days and hours are recognizable, variations – e.g. over weeks – are harder to discern.

To overcome this limitation, they use cluster analysis combined with a calendar-based visualization. Each cluster is visualized as a graph, while the days corresponding to the cluster are highlighted in a calendar (see figure 2.4). This allows users to identify day patterns and how they are distributed over the year and over the week.

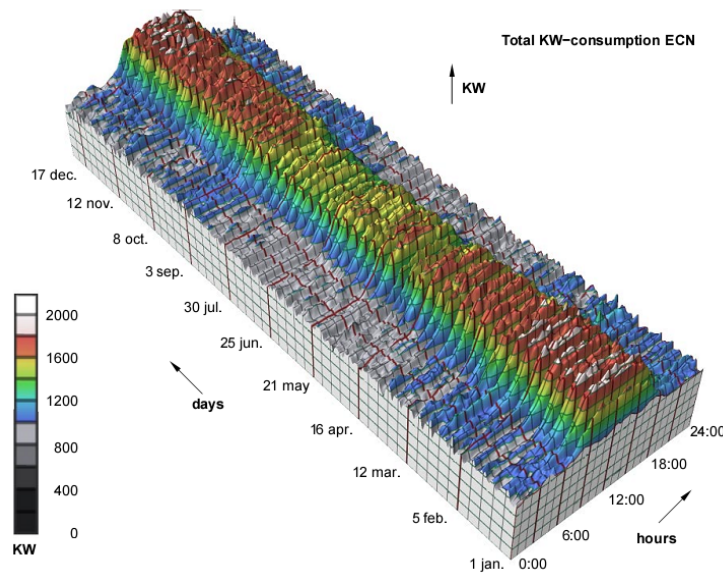


Figure 2.3. Power-demand displayed as a function of hours and days

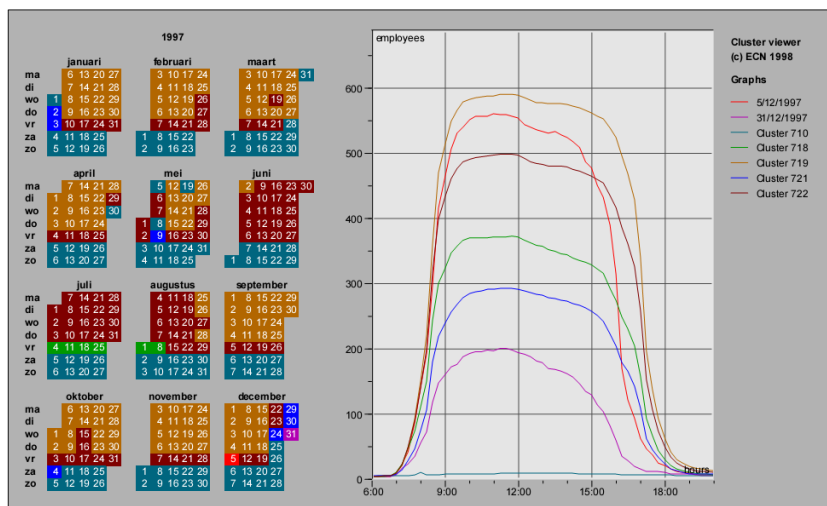


Figure 2.4. Clustered calendar view of the number of employees

2.2. Music Listening Histories

Music is a part of our lives – everyone listens to music. With that in mind, it is not surprising to see various research projects dedicated to music listening histories. The following chapters first describe various existing visualizations for music listening histories, and a work in the field of music psychology analyzing how people interact with music.

2.2.1. Visualization

A number of visualizations exist for music listening histories. While most of them work on a level of artists or genres instead of tracks, they demonstrate some of the kinds of insight that can be generated from music listening histories.

Strings and Tangle

In „Pulling Strings from a Tangle“ [11], Baur and Butz created two interactive visualizations for music listening histories: the first – called „Tangle“ – consists of a node-link diagram of all unique songs in the listening history, with sequences of tracks in the listening history represented by edges (see figure 2.5). Using a force-directed layout, the resulting visualization looks like a tangle of songs and mostly provides a global overview over a listening history. To analyze relations between songs as they appear in the listening history, users can use interaction mechanisms like panning, zooming and dragging nodes for closer inspection.



Figure 2.5. Tangle visualization

The second visualization – called „Strings“ – provides a session-based view of a listening history in a temporal context, where each session is displayed chronologically sorted as a horizontal string of songs (see figure 2.6). Identical songs in different listening sessions are connected by semi-transparent wide lines and users can again explore the visualization interactively.

In addition to analyzing listening histories, the visualizations can both be used to create playlists based on a map metaphor: by selecting start and end points, the application follows existing edges between them in order to create a playlist. The resulting playlists reflect the user’s listening behavior represented by the existing edges. On top of analyzation, this makes for an interesting secondary use case for the visualization.

As the authors note, the main limitation of the two visualizations is scalability, both in terms of the visualization concept and implementation. Both visualizations lose expressiveness with as few as a couple of hundred songs, and by using Java and the prefuse framework¹, the authors were only able to display up to 1.000 songs without losing a tolerable framerate. With growing music libraries and listening histories, these pose severe limitations for the visualization.



Figure 2.6. Strings visualization

LoomFM

LoomFM [12] is an interactive track-based visualization of two listening histories on a timeline, where one history is displayed above and the other below the timeline (see figure 2.7). By connecting multiple instances of tracks in the listening history with arcs, the visualization highlights sequential aspects of a listening history such as repetitions and differences in taste over time. Additionally, artist names and genres are displayed beyond the user timelines with their size corresponding to the number of consecutive songs sharing the artist or genre, which allows for observing general trends when zooming out.

The visualization allows interesting insights into listening histories such as the consumption intervals of songs, and explores the comparison of listening histories of multiple users. However, according to the authors „LoomFM suffers from low performance when displaying more than a thousand songs, an issue that comes up with realistically sized collections“ [12].

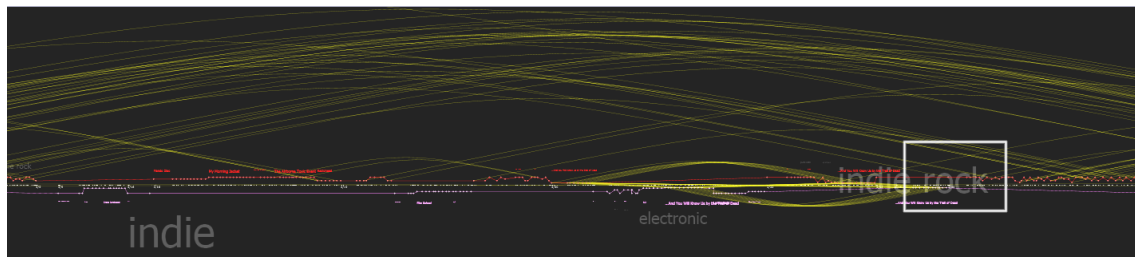


Figure 2.7. LoomFM visualization

¹ <http://prefuse.org>

Stacked Graphs

ThemeRiver [13] and Stacked Graphs [14] describe a visualization techniques for multiple time series data. Each time series is shown as a colored layer on the timeline using a layout method in which layers are stacked in a symmetrical shape with the x-axis at the center. The width of each layer varies with the time series' data points using a smooth interpolation from discrete data, creating aesthetically appealing graphs.

Figure 2.8 shows an example of such a graph from LastGraph [13], an online service for exploring Last.fm listening histories. The graph shows each artist from a user's listening history as a layer in the graph, making it possible to see the artists coming and going like „waves“. A similar graph can be generated by Last.fm's Listening Trends visualization [16] for the top 40 artists of any one or more Last.fm users. Figure 2.9 shows a graph of the listening histories from two users. Both graphs are made available as PDF documents and are thus non-interactive.

The choice of focusing on artists only makes the visualizations very readable due to a low number of dimensions, but at the same time only allows a very high-level overview over a listening history.

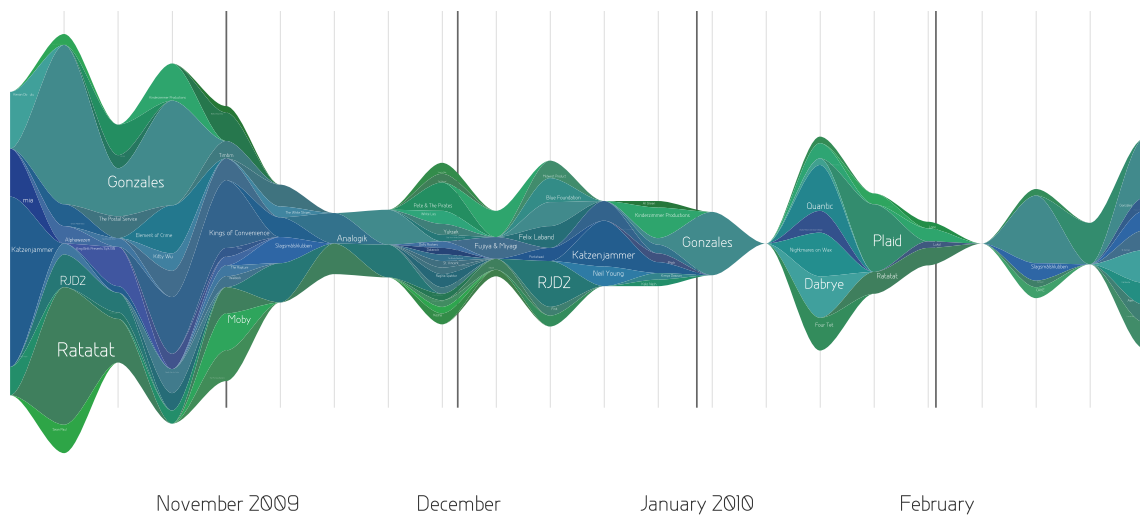


Figure 2.8. Stacked graph poster from LastGraph

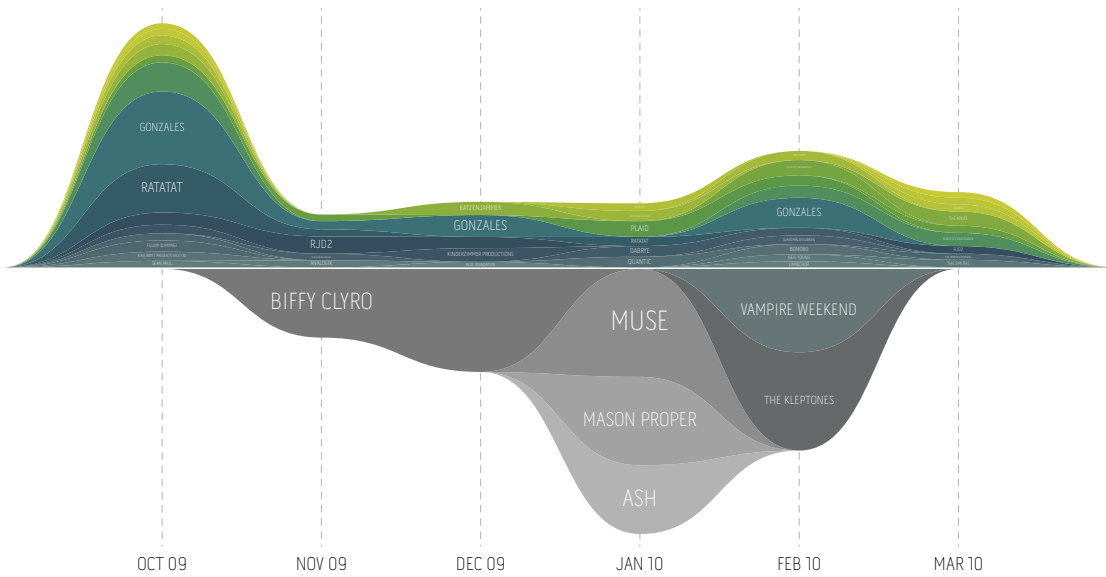


Figure 2.9. Graph of top artists for two users from Last.fm's Listening Trends

Tube Tags

Similar to stacked graphs, Last.fm's Tube Tags visualization [14] shows the changes in a listening history, but is based on the social tags (mostly referring to genres) of the songs instead of the artists:

„Each coloured line represents a tag, with key artists listed underneath. The white circles appear with tag names whenever there's an increase in listening for that tag, so you can quickly read the bias for that month.“ [14]

The visualization is again offered as a non-interactive PDF document and allows a very high-level overview over the predominant tags (or genres) of a listening history on a resolution of months (see figure 2.10).

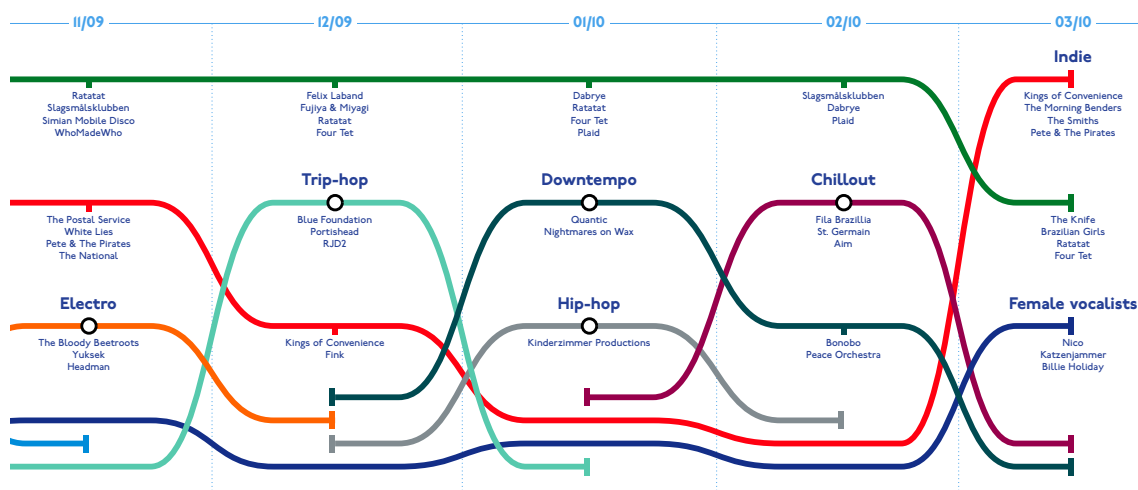


Figure 2.10. Last.fm Tube Tags visualization for a listening history

Last.fm Explorer

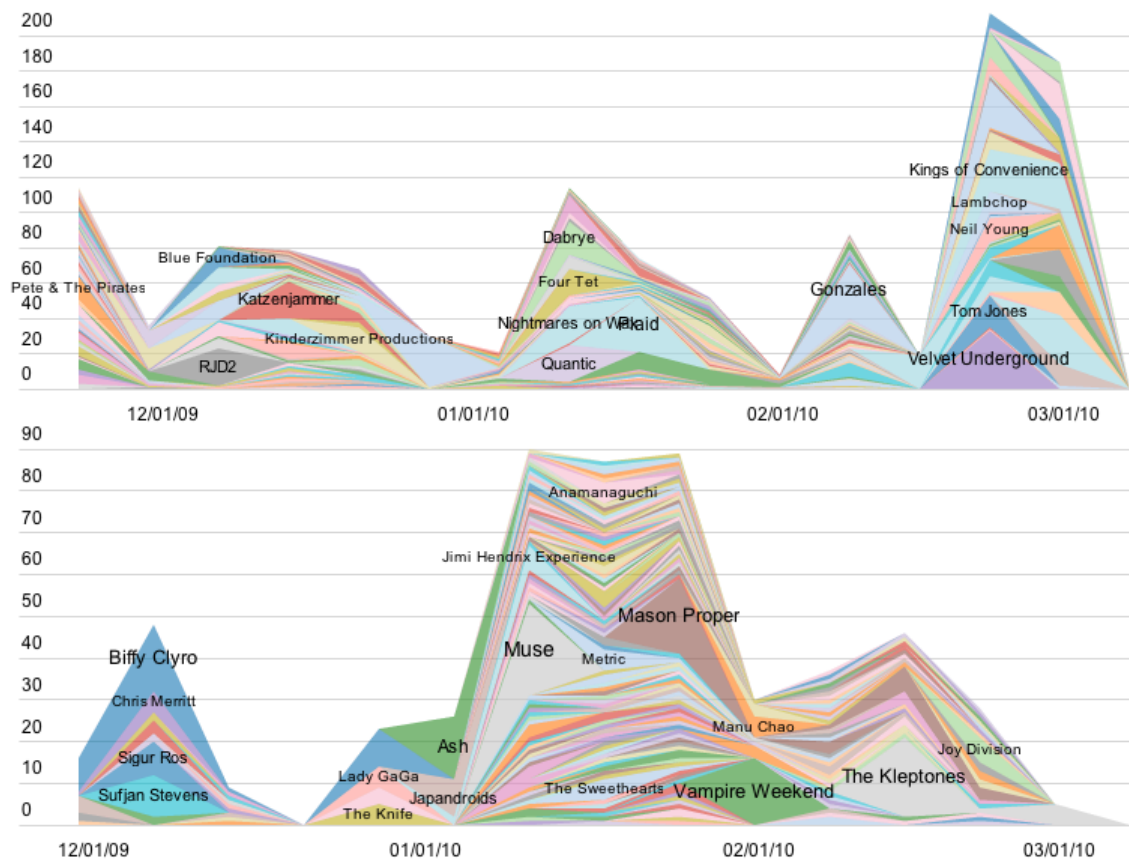


Figure 2.11. Visualization of two user's listening histories by Last.fm Explorer

Last.fm Explorer [15] is a interactive web-based visualization of music listening histories, allowing users to explore their Last.fm profiles using either a stacked graph or line graph that shows the top tags, artists, or tracks for every week. Users can apply filters and drill down into the listening history by double-clicking a tag or artist, which in turn shows a sub-set of the data in the next lower level of the hierarchy of tags – artists – tracks. Additionally, users can add a visualization of a second user's listening history, allowing direct comparison of two users' listening habits (see figure 2.11).

While the visualization is initially similar to LastGraph or Last.fm's Listening Trends, it shows how interactivity provides for a much richer experience and allows insights on a level of detail that would not be possible with static visualizations due to the amount of data examined. The biggest limitation lies in the data resolution of the visualization: the entries in a listening history are combined into one data entry per dimension for every week, due to which one loses the ability to get insights on a level of days or hours. Also, when choosing track-level detail in Last.fm Explorer, the visualization quickly becomes unreadable due to the number of tracks listened to during the course of a week.

Scrobbling Timeline

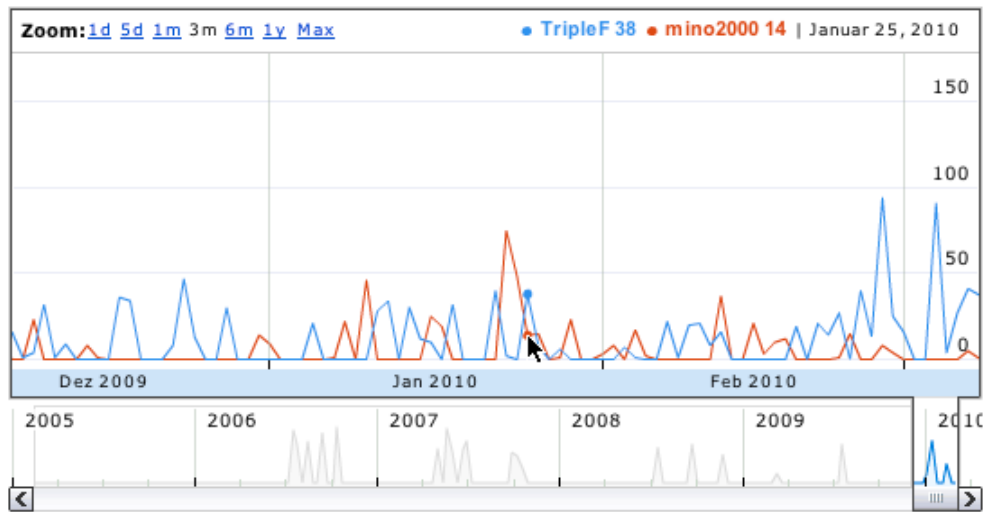


Figure 2.12. Last.fm Scrobbling Timeline visualization for two listening histories

Last.fm's Scrobbling Timeline [12] is yet another visualization offered by Last.fm, which allows users to investigate the scrobbling frequency for one or two listening histories, either as a whole or focused on a specific artist. For each day it shows the number of new entries in the listening histories, or the total number of entries up to that day by enabling a „cumulative“ mode (see figure 2.12). The visualization is interactive in the way that it allows scrubbing over data points to see the exact number of history entries for the highlighted day.

Again, the limitation here is the level of detail, as it does not show any patterns in the listening history pertaining to different artists or tracks.

Normalisr

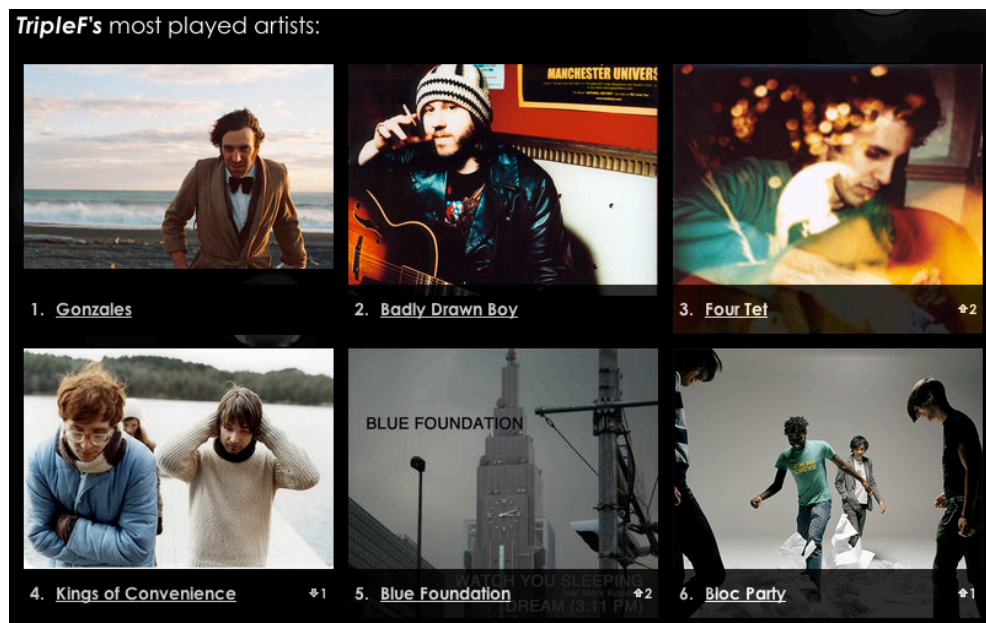


Figure 2.13. Artist chart by Normalisr

Similar to how Last.fm shows a user's favorite artists on every profile page, Normalisr [15] shows tables of a user's favorite artists and albums, either as a list or as visually appealing thumbnails. Normalisr's innovation lies in the calculation of the favorite artists and albums: instead of simply adding the number of times an artist or album was scrobbled, Normalisr bases the charts on the amount of time that was spent listening to the artist or album. The process for creating the charts is described as follows in the Normalisr Help [17]:

1. Takes the your last.fm username and grabs the XML list of your top 50 artists/albums;
2. Goes through those artists and grabs album and track data for them from the MusicBrainz web service;
3. Calculates the median track duration for each artist, using it to estimate how much time you have spent listening to the artist;
4. Sorts the artist list by estimated time.

According to the author, this results in charts that more accurately represent listening behavior when listening to artists with different track lengths, as artists with shorter tracks (resulting in more scrobbles in the same amount of listening time) are no longer preferred in the charts [18]. Figure 2.13 shows an artist chart in thumbnail mode by Normalisr.

While Normalisr provides for an interesting overview of one's listening habit, it fails to give any insight into changes in habits or in dimensions other than artists or albums.

2.2.2. Analysis

The field of music psychology analyzes how people interact with music. An interesting aspect in this regard is how music is used in everyday life, which was investigated by North et al. in a 14-day study with 346 participants [16]. The purpose of the study was to identify everyday uses of music in a natural setting: each participant was sent a text message on their mobile phone at various times of the day, upon which they had to complete a questionnaire about the music they had heard since the previous message. Using the questionnaires, five aspects of music listening were investigated:

1. Who do people listen to music with?
2. What do they listen to?
3. When do they listen to music?
4. Where do they listen to music?
5. Why do they listen to music?

An interesting finding of the study was that participants selected different genres depending on the purpose, if they had the choice of selecting the music. For example, music heard at leisure time was chosen for enjoyment, whereas music heard during work was another genre and had more practical reasons for listening, such as enhancing concentration. The results indicated that the voluntary and active use of music (in contrast to involuntary music listening, e.g. in shops) differed depending on the context.

With the time-consuming approach of having participants fill out questionnaires every day, the breadth of the study was limited. It stands to reason that using music listening histories could support such studies in the future, as both the „what“ and „when“ aspects of music listening are automatically recorded in the histories. Currently, such work would however be limited by missing tools for analyzing music listening histories.

2.3. Media Browsing Interfaces and Concepts

While music listening histories focus on the sequential list of tracks as they were listened to by a user, they can also be seen as a music library „in disguise“ since they usually contain a large percentage of a user’s music library (i.e. every track in the music library that the user listened to at least once). With this in mind, this chapter takes a look at interfaces and concepts produced by the HCI and information retrieval fields for browsing media libraries for photos and music.

Time Quilt

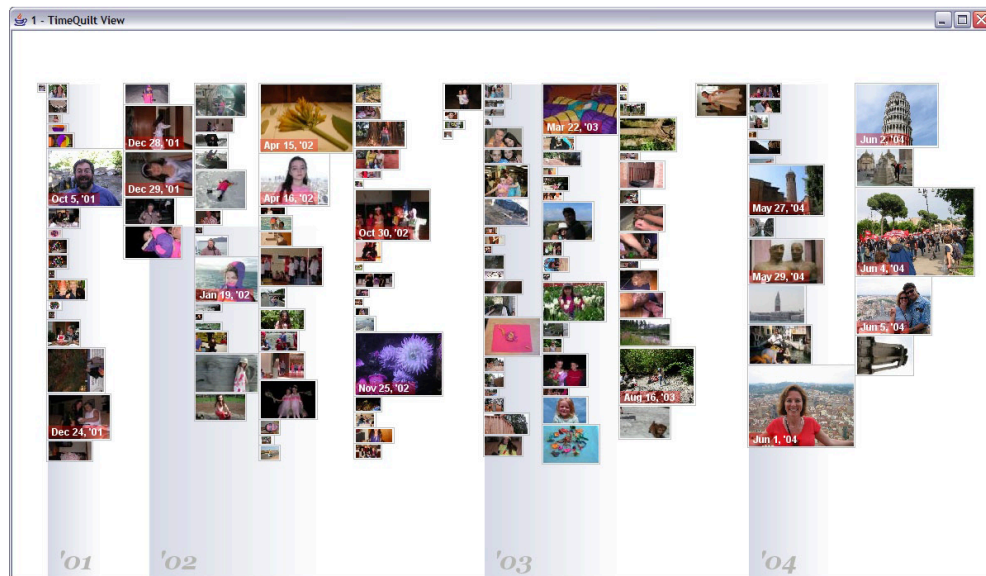


Figure 2.14. Time Quilt screenshot

Time Quilt [17] is a photo browser that presents photos on a timeline in a space-efficient way while retaining temporal order. All photos are first clustered by their creation dates, and representative thumbnails are used when zoomed out to present a cluster of images instead of presenting all images of a cluster at a smaller size. The items in the timeline are then wrapped into vertical columns to achieve better overall use of a rectangular screen space (see figure 2.14).

According to a study, using the Time Quilt layout users were able to find photos in a large photo collection 45% faster than using a standard timeline.

PhotoMesa



Figure 2.15. PhotoMesa screenshot

Similar to Time Quilt, the PhotoMesa [18] image browser presented by Bederson aims at showing large photo collections with maximum usage of screen space using a zoomable user interface (ZUI) originally developed by Perlin et al. [19]. Photos are grouped by directories or other available metadata, and the groups are laid out in a 2D space-filling manner using one of two algorithms presented by the author (see figure 2.15). Users are able to zoom multiple levels into groups of images by clicking them with the mouse until the point where a single image is shown, and zoom out again using a right-click or by pressing a key on the keyboard. Additionally, images are enlarged when leaving the cursor on one of them for a short time.

While the temporal order is not retained in the same way as in Time Quilt, the author highlights the zooming interface as a way to quickly navigate within a photo collection. The zooming is performed in an animated way, which according to the author makes it easier to follow the interaction.

Calendar Browser

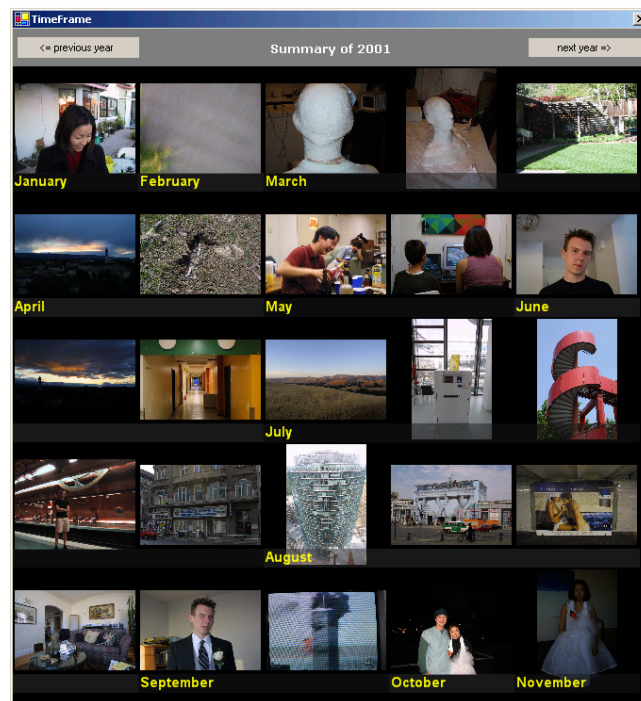


Figure 2.16. Calendar Browser screenshot in year view

Calendar Browser [20] is an application for browsing photos using a calendar metaphor at different time granularities like years or months. For example, when selecting the year granularity, all photos shown in the display panel will have been taken during the same year (see figure 2.16). Calendar Browser clusters images recursively by time at increasingly fine granularity, assuming that a photographer takes pictures in „bursts“. This allows organizing images by photographic themes: for example, one cluster might be a birthday party, consisting of sub-clusters like „friends arriving“ and „opening presents“. The authors call this process „time-based summarization“ and use it to assign images representative of a cluster to the slots available in the calendar view.

In a user study, the authors were able to show a 33% improvement in speed for finding specific images compared to a traditional image browser. The work thereby shows how utilizing the time dimension can lead to significant improvements.

Musicream

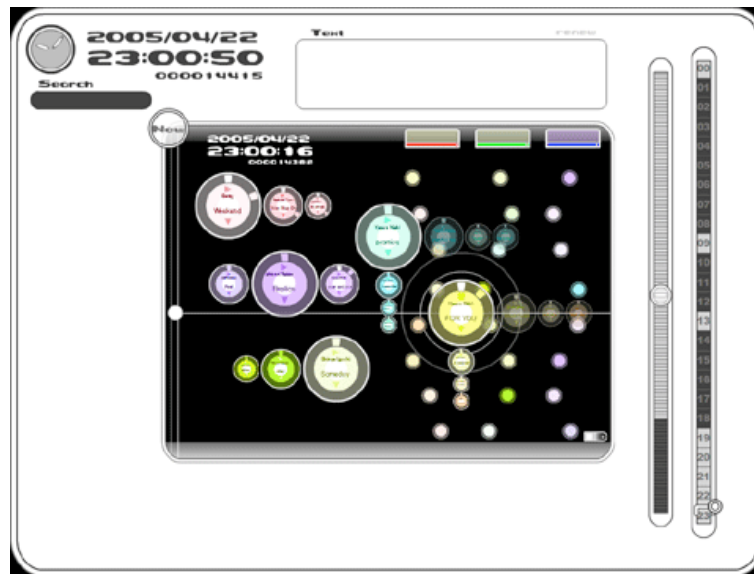


Figure 2.17. Musicream screenshot in time machine mode

Musicream is a novel music player that helps users discover music and generate playlists using audio-based similarity measures [21]. Streams of music pieces are represented as discs in the visualization, where each disc describes a different mood. Users can pick out a piece to listen to or „collect“ similar pieces by dragging a seed song into one of the streams.

Among its features, Musicream also employs a „time machine“ mode: by recording all user interactions, the software allows reproducing past applications states and thereby recalling the past. This can be used to undo any actions, but also to search the playback history for other times when listening to a specific song, to perform a „past similarity search“ to find musical pieces from the history similar in mood to a given starting piece, or simply to access an old playlist (without the need to actively save playlists). Past states can be accessed using two sliders that enable moving through the recorded history by either units of seconds relative to the present or by dates and time (see figure 2.17).

While the „time machine“ concept collects information similar to listening histories, it goes beyond them by saving the entire application state for every point in time, thereby enabling further features like accessing past playlists even if they were not played in their entirety. On the other hand, much of the functionality provided could also be recreated using listening histories, which suggests a potential use case for music listening histories.

3. Conceptual Design

After highlighting related work, the following chapters first describe the underlying data that can be analyzed through the visualization, before describing the design principles and the overall interface- and interaction design of the application.

3.1. Music Listening Histories

Music listening histories (or just „listening histories“) such as the ones available by Last.fm are created by recording each instance of a single person (i.e. the *history owner*) listening to music. This is accomplished by installing a daemon service running in the background on the person's PC, which in turn keeps track of the music played through various media player applications such as Apple's iTunes as well as mobile devices like Apple's iPod, and transmits information about the tracks as well as a timestamp to Last.fm. This process is called „scrobbling“ by Last.fm [24].

In result, music listening histories consist of a finite set of triplets (listening history entries), each containing the following information:

- Timestamp
- Artist
- Title

This information makes up the user profiles made available for every user by Last.fm through its website and web services. The artist and title describe the song (or track) the history owner listened to, and the timestamp describes the date and time when the song was played. It is important to understand that multiple history entries might refer to the same song, i.e. when the same song gets played multiple times throughout the history.

Listening histories are not perfect though, in the sense that they usually do not contain every piece of music the history owner listens to (data gaps), and additionally might contain entries for songs that the history owner did not listen to (noise). While scrobbling software is available for a range of media players and also portable devices like Apple's iPod, other devices might not be supported. As such, data gaps can stem from various situations: the user uses a media player or device not supported by the scrobber, the user does not have the scrobbling software installed on the computer he is using, the user listens to music through websites that cannot be tracked by Last.fm, or the user uses other means to listen to music such as CDs, vinyl records, or AM/FM radio that are not trackable through software. Moreover, the user might be listening to music in a way that is not under his control, such as by visiting concerts, parties, or clubs where music is played. Noise on the other hand can stem from situations such as another person using the same computer for listening to music, or the user does not stop the music from playing when leaving the computer.

Last.fm uses the collected listening histories for collaborative filtering [22] in order to recommend artists and songs to its users based on their Last.fm profiles. In this use case, data gaps and noise as described above are less of an issue as collaborative filtering algorithms do not have to rely on exact data and are sufficiently fuzzy for noise to be ignored. For accessing and analyzing listening histories however, these issues need to be kept in mind as they lower the quality of the data being analyzed.

From the information about the song (the artist and title), related information can be determined using web services available from Last.fm:

- the album the song might be part of
- an image of the album artwork
- user-created tags („social tags“)

The social tags are used in the visualization to infer the genre of the track. For this purpose, a mapping from tags to genres was created. For each song, the tag with the highest tag count available in the mapping is used to determine the genre. The mapping is described in more detail in [Chapter 5.4](#).

3.2. Design Principles

After analyzing the related work described in [Chapter 2](#), two design principles emerged for maximizing the insight facilitated by the visualization:

- Embrace time
- Enable interactivity

The first design principle – embracing time – follows from listening histories being time-oriented data. As noted by Aigner et al. in 2007, „understanding temporal relations enables us to learn from the past to predict, plan, and build the future“ [23]. Hence, considering the time dimension appropriately in the visualization allows comparing data located at different points in time, thereby detecting trends and patterns leading to insight and understanding of the data. As shown in [the following chapter](#), the application visualizes the time dimension of the listening histories on two different levels of temporal aggregation, namely days and time of day.

Comparing the non-interactive visualizations described above to the interactive ones, it also quickly becomes clear how interactivity enables a much broader range of insights, leading to the second design principle. As noted in Aigner et al.’s framework for visual analytics, „interaction methods and navigation are essential to explore the data as well as the parameter space“ [24]. The application therefore employs a broad range of interaction concepts to enable insight especially on the track level, which will be shown in [Chapter 3.5](#).

Finally, following the „infovis for the masses“ approach described in [6], two main categories of user tasks were identified for the visualization and realized in two modes for the visualization, each optimized for a different use case:

- **Analysis Mode**

This mode can be used for interactively analyzing an arbitrary listening history (not necessarily the user’s own history) in three basic dimensions: time, tracks and genre. Integrated tools like searching, playlist highlighting, mouse-over information, and zooming allow the user to find patterns and generate insight for the visualized music listening history in each of those dimensions.

- **Personal Mode**

This mode adapts the visualization shown in Analysis mode for exploring the user’s own past. By adding the user’s photos and calendar entries as landmarks to the visualization [7], the application can be used for reminiscing and storytelling. For example, it allows listening to the top tracks from the last vacation while watching a slide show of the corresponding photos, or finding the most influential tracks (i.e. the tracks listened to most during a specific timeframe) using the tracks highlighted in the visualization.

3.3. Visualization of the Music Listening History

For visualizing a music listening history, the time-based data is considered two-dimensional by splitting the time scale into days and time of day, which are then mapped onto different axes. The application thus visualizes the individual entries of a listening history as nodes on a two-dimensional timeline: along the x-axis are the days of the listening history in ascending order (with the past on the left and the present on the right), while the hours of each day are shown along the y-axis (in UTC/GMT, with midnight at the top and midnight of the next day on the bottom). Consequently, all history entries from a single day are shown along a vertical line.

The horizontal distance between consecutive days can be altered, thereby allowing users to zoom into the listening history to analyze several days in more detail, or zoom out to get an overview of the overall listening activity.

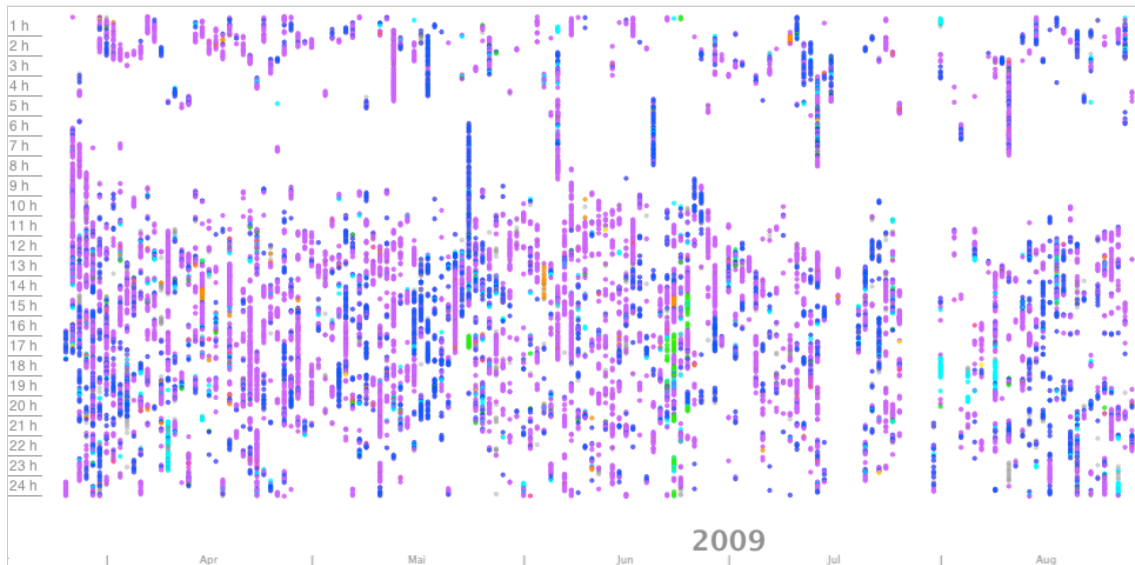


Figure 3.1. Screenshot of the visualization with labels for both axes

Figure 3.1 shows a screenshot of the visualization of a listening history. As can be seen, the x-axis is labelled with month names and the year, while the y-axis is labelled with hours. Each entry from the listening history is shown as a round node. The color of each node corresponds to the genre of the song. Following is a list of the node colors used in the visualization with their respective genre:

- **classical**: orange
- **jazz**: yellow
- **funk**: green
- **hip-hop**: turquoise
- **electronic**: blue
- **rock**: purple
- **metal**: pink
- **unknown**: grey

While there is no inherent connection between colors and genres, the colors were chosen for maximum visual distinction between each other.

By default, all nodes have the same size, giving equal visual importance to all entries in the listening history. In Personal Mode however, the node size varies according to the relative importance of the node (see [Chapter 4.2](#)).

3.4. Visualization of Personal Streams

In addition to visualizing a listening history, the application also allows users to optionally display other time-based „streams“ alongside the listening history visualization (enabled by default in Personal Mode). In the current version, this includes showing personal photos and calendar entries as „personal streams“, which are displayed just below the listening history. Figure 3.2 shows a section from a visualization including personal streams. The calendar entries are displayed below the photos as horizontal lines spanning the dates of the calendar entry. The different colors for the calendar entries correspond to different calendars the user has set up.

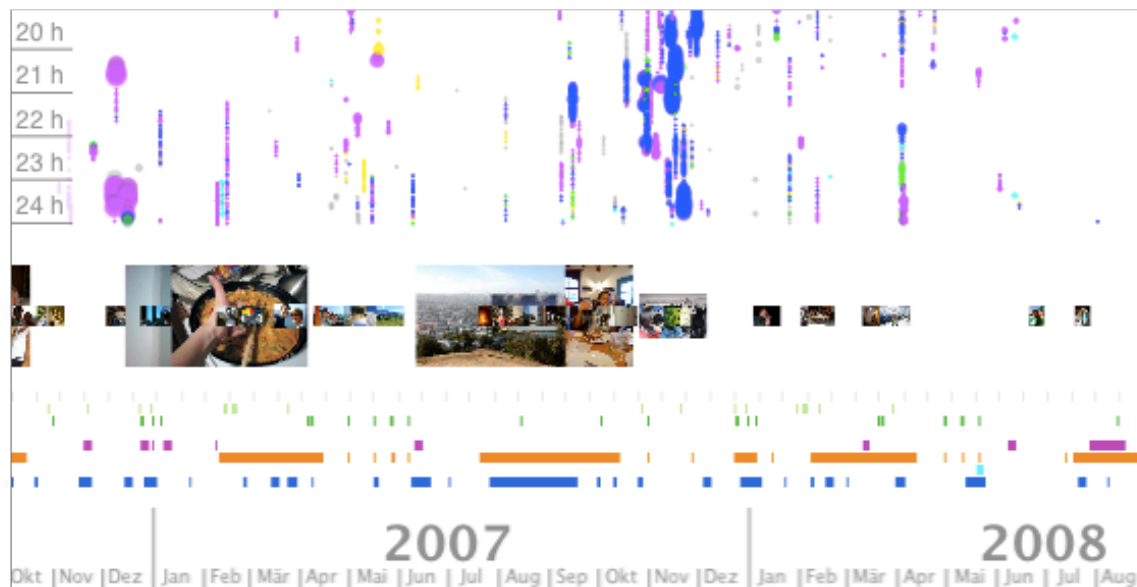


Figure 3.2. Screenshot (detail) of the visualization showing the photo and calendar streams

The personal streams are displayed in the following way:

- **Photo Stream**

The photo stream consists of all photo events from the user's iPhoto library. Each event is displayed with its key photo assigned to the event in iPhoto. The position of the photo in the timeline corresponds to the time period of the event (set by the event's start date and end date). The size of the photo corresponds to the number of listening history entries present in the listening history during the time period of the event. Hovering over an event magnifies the photo and allows scrubbing through all of the event's photos. Clicking an event shows a slide-show of the event's photos while playing the corresponding music from the listening history.

- **Calendar Stream**

The calendar stream consists of the all-day events from all local calendars in the user's iCal database. Each calendar entry is displayed as a horizontal line in the timeline, with the start and end position of the line corresponding to the start date and end date of the calendar entry. The entries' color corresponds to the color assigned to the calendar in iCal. Clicking a calendar entry plays the corresponding music from the listening history.

3.5. Interactive Usage

Using a mouse or a trackpad on a notebook computer, the user can navigate and explore the visualization interactively by pointing at and clicking elements in the visualization. The following is a description of the different ways of interacting with the visualization.

3.5.1. Navigating and Scaling the Timeline

Since the timeframe of the listening histories – and thereby the length of the timeline – varies between users, the application allows users to scroll the visualized timeline horizontally using two different methods of interaction:

- Clicking anywhere in the timeline and dragging the mouse while holding the mouse button moves the timeline by the amount the mouse is moved.
- When running the application on a notebook, the timeline can also be moved using the „two finger trackpad scrolling“ feature available on newer Mac notebooks [3].

In order to accommodate different needs regarding the level of detail the visualization should provide, the application lets the user scale the timeline using a slider control. Using this control, the user can effectively vary the horizontal space between the individual days, either moving them together to zoom out and get an overview, or spreading them out to look at a smaller timeframe in detail. Additionally, the scale can also be changed using a „pinch“ gesture on the trackpad of newer Mac notebooks.

Since the vertical timeframe (time of day) stays fixed, the application does not allow scrolling vertically but merely spreads out the nodes to fit the available space as defined by the vertical window content size. The size of the nodes can also be varied using controls available from the application menu. The default size has been chosen so that nodes can still be distinguished vertically (i.e. they mostly do not overlap) with song lengths of approximately 4 minutes and when maximizing the window using a screen size of 1280x1024 pixels.

3.5.2. Accessing Information about History Entries

The primary way of interacting with the visualization is by pointing at nodes with the mouse cursor. In order to allow investigating individual entries in the listening history, pointing at a node shows the following information about the history entry in a semi-transparent overlay window following the position of the mouse cursor:

- name of the song, artist, and album (if available)
- weekday, date, and time of the listening history entry (in GMT)
- total number of times the song was played in the listening history
- calculated weight of the node in the history (see [Chapter 4.2](#))
- genre
- tags with tag weights, ordered by descending tag weights

Similarly, pointing at photo events or calendar entries shows additional information about those entries.

To allow easier identification of the day and time of day the current mouse cursor position corresponds to, the visualization also shows crosshairs as thin light-grey lines, which mark the cursor position on the labels of the x- and y-axis. Figure 3.3 shows a screenshot of the overlay window and the crosshairs.

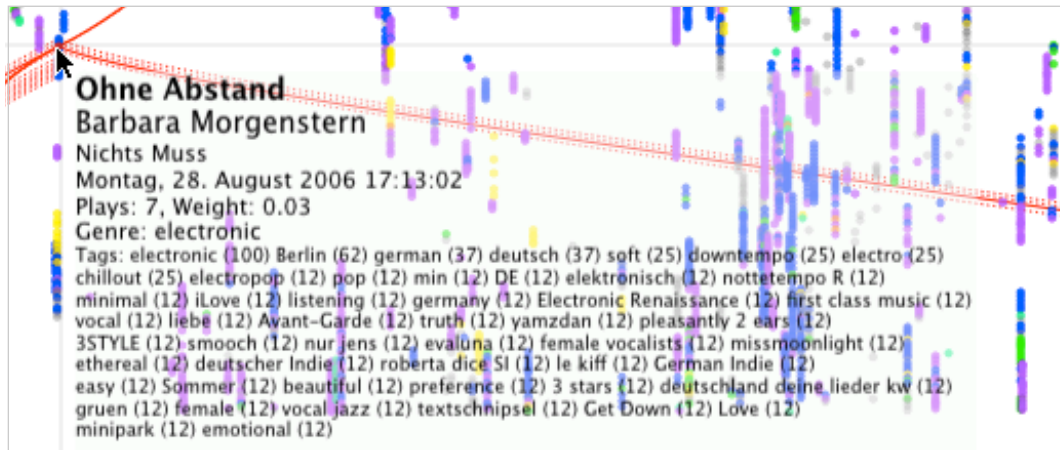


Figure 3.3. Screenshot showing the crosshairs and overlay window

In addition to providing information about specific entries, each visualization also shows the following statistics about the entire listening history:

- Total number of history entries
- Total number of unique tracks

When performing a search (see Chapter 3.5.5), the application additionally shows the number of history entries and tracks matching the query, as well as their percentage relative to the total number of history entries and tracks.

3.5.3. Highlighting Identical Tracks and Track Sequences

Hovering over a node will show appearances of the same track in the listening history connected through solid red lines. This allows to visually explore the temporal distribution of a single track in the listening history, e.g. when it was played first, how often it was played in total, the intervals between each occurrence of the track in the listening history, and possible clusters where the track was played more often in a shorter timeframe.

Additionally, the application detects other occurrences of the sequence of tracks the listening history entry is part of in the listening history, and shows these sequences connected through dashed red lines. This allows to easily see if the track was played as part of an album or playlist that was also played at other times in the listening history. Figure 3.4 shows a screenshot of a listening history with connections from a selected track to other occurrences of the same track and track sequences in the listening history.

Highlighting can be temporarily disabled by holding the shift key while moving the mouse.

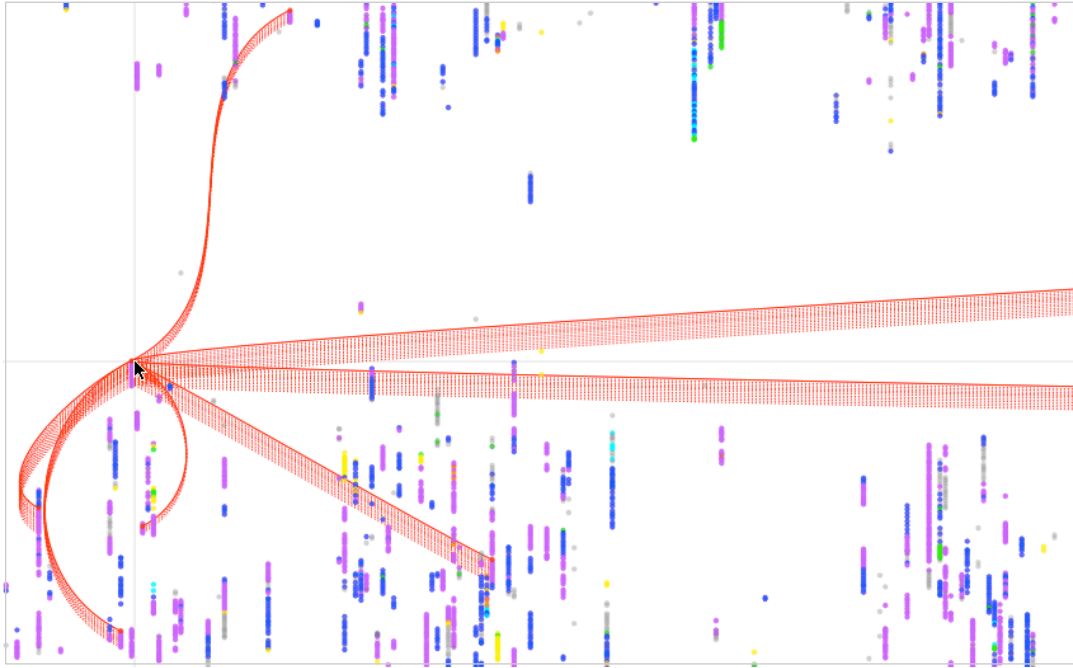


Figure 3.4. Screenshot with connections from the selected node to other occurrences of the track and track sequence in the listening history

3.5.4. Track Playback

Clicking a node plays the track from the user's iTunes library. If the track is not found, the node shows a „shake“ animation, moving the node slightly left and right several times in succession. Using controls available in the window's toolbar, the user can pause playback and skip to the previous and next track in the listening history. When using the skipping controls, the application automatically ignores tracks that cannot be found in the iTunes library, thereby playing the next or previous track that can be found in the iTunes library. Clicking outside a node stops playback.

The application also allows playing tracks corresponding to the time period of a photo event or calendar entry, or from a freely selected time period. For this, the user has to either click an entry in one of the personal streams, or click-and-drag within one of the axes' labels to select a specific timeframe (in the x-axis) or time of day (in the y-axis). Pressing the Ctrl-key while dragging, or dragging while pressing the right mouse button, allows selecting an arbitrary period using constraints for both the date and the time of day. User feedback is provided using a grey semi-transparent rectangle while dragging, switching to green when the tracks from the time period are being played. When playing the tracks from a photo event, the application additionally shows a slideshow of the event's photos.

By default, the application will play the tracks from the listening history in sequence when selecting a time period, starting with the first available track that falls within the selected period. By enabling the „Charts Mode“ from the application menu (enabled by default in Personal Mode), the application changes to playing the tracks in descending order of the number of occurrences of the track within the selected time period. This allows playing the most frequently played tracks and creating „charts“ for an event or other time period. The resulting playlist can be saved to iTunes using the corresponding command available in the application menu.

The application also provides a command that can be used to show the track's corresponding entry in the user's iTunes library. This can be used, for example, to create playlists in iTunes of tracks that were discovered through the visualization.

3.5.5. Searching

The search field in the upper-right corner of the window allows searching the listening history by time, tracks and genre. This allows for a more in-depth inspection of a listening history for specific patterns. Following are some examples of possible search terms:

- *13*: All history entries between 13:00 and 13:59 GMT.
- *1-6*: All history entries between 01:00 and 06:50 GMT.
- *Sa* or *Saturday*: All history entries on Saturdays.
- *Mo-Fr*: All history entries between Monday and Friday.
- *Apr* or *April*: All history entries in April.
- *Jul-Aug*: All history entries between July and August.
- *2009*: All history entries in 2010.
- *2009-2010*: All history entries between 2009 and 2010.
- *Rock*: All history entries with the genre or tag „Rock“ or the name of the artist or album containing „Rock“.
- *Beatles*: All history entries with the genre or tag „Beatles“ or the name of the artist or album containing „Beatles“.

When entering search terms, each term is displayed as a separate token. Multiple search tokens can be entered by pressing enter or comma after each term. Using a menu attached to each token, it is possible to select a specific category to search in (genre, artist, title, album, or tags) or to invert the search for the token by selecting „NOT“. Multiple date or time search tokens are combined with disjunctions („or“), while multiple text search tokens as well as time and text search tokens are combined with conjunctions („and“). Weekday names have to be entered in the system language.

Figure 3.5 shows a screenshot of the search field containing two tokens, and showing the menu for the second token.

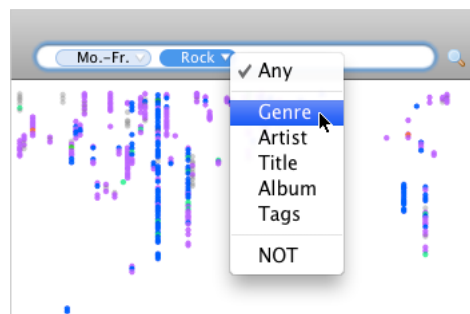


Figure 3.5. Screenshot of the search field

4. Application Operation

The following sections describe how the different application modes – the Analysis Mode and the Personal Mode – can be used for different tasks and to generate different kinds of insight.

4.1. Analysis Mode

The „Analysis Mode“ allows for the interactive visual and analytic inspection of music listening histories in three basic dimensions:

- **Time:** time of day, date
- **Tracks:** artists, titles, playlists
- **Genre**

Using the visualization with the integrated tools like searching, playlist highlighting, mouse-over information, and zooming, a user can generate insight for the visualized music listening history in each of those dimensions. The following sections describe the kind of insight the visualization allows.

4.1.1. Insights in the Time Dimension

Time can be seen as the most prominently visualized dimension in the visualization. To recall, along the x-axis are the days of the listening history, with the first entry of the history being on the left („the past“) and the most recent one on the right („the present“). The bounded y-axis marks the time of day of the history entries for each day on the x-axis, ranging from midnight at the top of the view, to just before midnight of the following day at the bottom.

All indicated times in the visualization are using UTC, i.e. the time of day is relative to the UTC±0 time zone and does not change with a change of seasons.

The following insight can be gained in the time dimension:

- **Amount of music listening**
The overall density of nodes in the visualization shows the amount of music the owner of the history listens to on average.
- **Predominant time of day for music listening**
Most histories show an increased average node density during certain times of each day (along the y-axis), in contrast to a sparse node distribution or no nodes during the remaining times. This indicates the prevailing time of the day the history owner uses for listening to music, which can also vary over time or between different days of the week (e.g. weekdays vs. weekends).
- **Sleep-wake rhythm**
If a history shows a very sparse or nonexistent node distribution during a certain 6-9 hour period of the day, this usually indicates the history owner's time of the day used for sleeping.
- **Regularity of the daily routine**
Gauging by the distinctness of the „sleeping period“ and other prevalent changes in node density during certain times of each day in the visualization (e.g. lunch breaks, dinner, etc.), one can estimate the regularity of the daily routine of the history owner. For example, while one history might have a clearly visible break between 12:00 and 13:00 indicating a regular lunch break, another history might only show a very blurry decrease in node density around the same time, indicating that the history owner varies his or her lunch break time.
- **Listening variations between days of the week (e.g. weekdays vs. weekends)**
Using the search feature, a user might be able to show a variation in listening habit between different times of a week. For example, a history owner might start listening to music later during weekends than during the week.

- **Multi-day events (e.g. holidays)**

A history might also show multiple consecutive days with an abrupt change in node density, indicating a special event in the history owner's life (e.g. a holiday). For example, a history might show a two week area with no nodes in the month of August, which could correspond to a holiday where the history owner travelled without listening to his or her personal music collection or without using a Last.fm-compatible music listening device.

- **Indication of the history owner's time zone**

Using the time of the „sleeping period“ in the history, one can estimate the time zone the user lives in by applying knowledge about predominant sleeping periods in different time zones.

- **Relocation with change in time zone**

If a history shows a sudden shift in the daily routine by one or more hours, this can indicate that the history owner temporarily or permanently relocated to another time zone.

- **Length of listening sessions**

The vertical length of continuous sequences of nodes indicates the length of listening sessions.

4.1.2. Insights in the Tracks Dimension

While time is encoded very directly in the visualization, the individual tracks in the history and their associated information like artist and album names as well as detection of playlists (albums or user-generated playlists) can only be seen by (inter-)actively inspecting a listening history using the mouse: pointing at an individual node in the visualization will show all available information about the connected track (title, artist, album, genre, tags, and number of plays), as well as highlighting connections to any other plays of the playlist that the track might be part of in the history.

Additionally, the program shows the total number of both history entries and unique tracks in the status bar.

The following insight can then be gained regarding the tracks:

- **Relation of tracks to history entries**

The total number of unique tracks in the history in relation to the total number of history entries as indicated in the status bar gives a hint as to whether the history owner focuses on listening to a limited number of tracks repeatedly, or varies his listening habits more across a larger number of tracks. This can also be related to the way music is listened to: for example, listening to internet radio stations or personal radio stations as offered by Last.fm typically produces a larger number of tracks in relation to the number of history entries, as the tracks would vary more and often only be played once.

- **Listener type**

Using the playlist detection, it can be determined if the history owner mostly listens to albums or playlists, or if he prefers listening to random tracks (shuffle mode) or internet radio stations. As an album- or playlist-centric listener often produces the same or similar sequences of tracks in the history multiple times, inspecting such a history will mostly show highlighted playlists consisting of multiple tracks when inspecting the visualization with the mouse. On the other hand, a history of a listener who mostly plays tracks randomly might show connections to multiple instances of the same track in the history, but as these tracks would most likely be surrounded by different tracks in the timeline at each instance, no playlist would be detected and highlighted in this case.

- **Variability of tracks**

By highlighting multiple occurrences of the same track in the history when pointing at a node with the mouse, the visualization shows how much the history owner listens to the same tracks versus regularly playing new tracks not occurring in the history before.

- **Locality of tracks**

Using the highlighting of multiple occurrences of identical tracks in the history, it can be determined if the history owner mostly listens to the same tracks multiple times within a shorter timeframe and then „forgets“ about them (and moves on to new tracks), or if he mostly sticks to a set number of tracks that are listened to again regularly throughout the history. Another type of user might listen to music in phases or „waves“, where the same music gets played multiple times for a couple of weeks or months, before interest in the music ceases for a while and then returns [25].

- **Rediscovery of tracks**

Similar to the locality of tracks, it can be seen from the patterns of multiple occurrences of identical tracks in the history if a history owner often rediscovers music listened to in the past.

- **Length of tracks**

By looking at the spacing between continuous sequences of nodes in the visualization, the length of the tracks in the sequence can be roughly estimated: a sequence with longer distances between the nodes could indicate longer tracks (e.g. classical music), while a short spacing usually indicates shorter tracks. However, skipping and pausing tracks can lead to shorter or longer node distances independent of the length of the tracks.

4.1.3. Insights in the Genre Dimension

Finally, the genre of tracks is visually encoded in the visualization by means of different node colors. Investigating the color pattern, the following insights can be gained regarding the genre of the tracks in the listening history:

- **Predominant genre**

Looking at the overall color distribution, one can determine the predominant genre in the listening history as the predominant color of the nodes. This would be especially well-defined for listening histories where the history owner has a clear preference for one genre. For other listening histories with a larger variation of music from different genres, the predominant genre might be less clear or there might even be multiple equally dominant genres.

- **Change of genre over time**

Adding the time axis to the picture, it can be seen if the predominant genre changed over time. For example, a history might show mostly rock music (purple) in the beginning of one year, but more electronic music (blue) at the end of the next year. In between, there would either be a continuous change from purple to blue, or a history owner might have changed his taste more abruptly as signaled by a sudden change in average node color.

- **Variability of genre**

By looking again at the overall node color pattern, it can be determined if a user has a very strong preference for one genre with little or no other genres in between, or if there might be other genres spread throughout the history, either continuously or as „bursts“ of multiple nodes with a different color.

4.2. Personal Mode

In „Personal Mode“, the application adapts the visualization shown in Analysis Mode for inspecting a user's own listening history for the purposes of reminiscing and storytelling. In this regard, adding personal information helps to make sense of discerned patterns in the context of a person's life, and also allows connecting events with the music listened to during these events.

As described in [Chapter 1](#), linking a time of one's life with the relevant music helps other related memories to also reappear. However, not every song we heard in the past is equally well-suited to work as a memory cue and reminds us of a specific time in our lives. Memory theory says that the more contexts a particular cue is associated with, the less effective it will be at bringing up a particular memory [26]. While some songs we might only hear for a short time

e.g. during a summer vacation, there are often favorites in our music collections that we listen to more regularly and which are thus less likely to be associated with a specific context. For the personal use case, the visualization therefore highlights the songs that were listened to more intensely during certain times in the listening history, and also allows playing the most-listened songs (charts) for specific timeframes. Finally, an integrated photo slideshow makes it possible to relive the past with both visual and musical cues.

Accordingly, the Personal Mode differs from the Analysis Mode in three ways:

- **Weighted history entry nodes**

In contrast to the Analysis Mode, where all history entry nodes in the visualization have the same size and thus equal visual importance, the Personal Mode resizes history entry nodes according to their significance in the listening history. This is done by calculating a normalized weight based on the number of times the associated track was played in a 30 day period surrounding the history entry (see [Chapter 5.4.2](#)). The visualization thereby allows the user to identify tracks that bear a stronger meaning for a certain time period in order to use them as a means of reminiscing about that period.

- **Additional event streams: photos and calendar entries**

Using the additional event streams, the user can listen to the music played during events simply by clicking on them, and in the case of the photo stream, the application also shows a slideshow of the photos to the playing music.

- **Charts playback mode**

When playing history entries for an event (e.g. for photos or calendar entries), the songs are not played in the order of the entries in the listening history, but in the descending order of the number of plays of the songs during the selected timeframe.

In summary, although most of the same insight as in Analysis Mode can be generated, the aim of this use case is not the analytic inspection of a listening history, but to make one's own history accessible so it can be used for reminiscing and storytelling, as well as for finding music that was once important during a specific time in one's life.

4.3. Examples of Music Listening Histories

The following figures show screenshots of the application visualizing a variety of music listening histories from different Last.fm users around the world. The scale of the timeline and the vertical window size is the same for all screenshots, which makes it possible to compare the different listening behaviors in the time dimension. Each screenshot is annotated with the country from the user's Last.fm profile.

While the tracks dimension only becomes accessible through interacting with the visualization, a number of insights can be readily observed from these screenshots in the time dimension:

- Amount of music listening
- Predominant time of day for music listening
- Sleep-wake rhythm and regularity of the daily routine
- Multi-day events
- Indication of the history owner's time zone

The remaining insights in the time dimension are either not clearly visible in these particular listening histories (e.g. relocation with change in time zone), or would require a more detailed analysis of the visualization by zooming the timeline (e.g. listening variations between days of the week).

In the genre dimension, all of the insights listed above can be observed:

- Predominant genre
- Change of genre over time
- Variability of genre

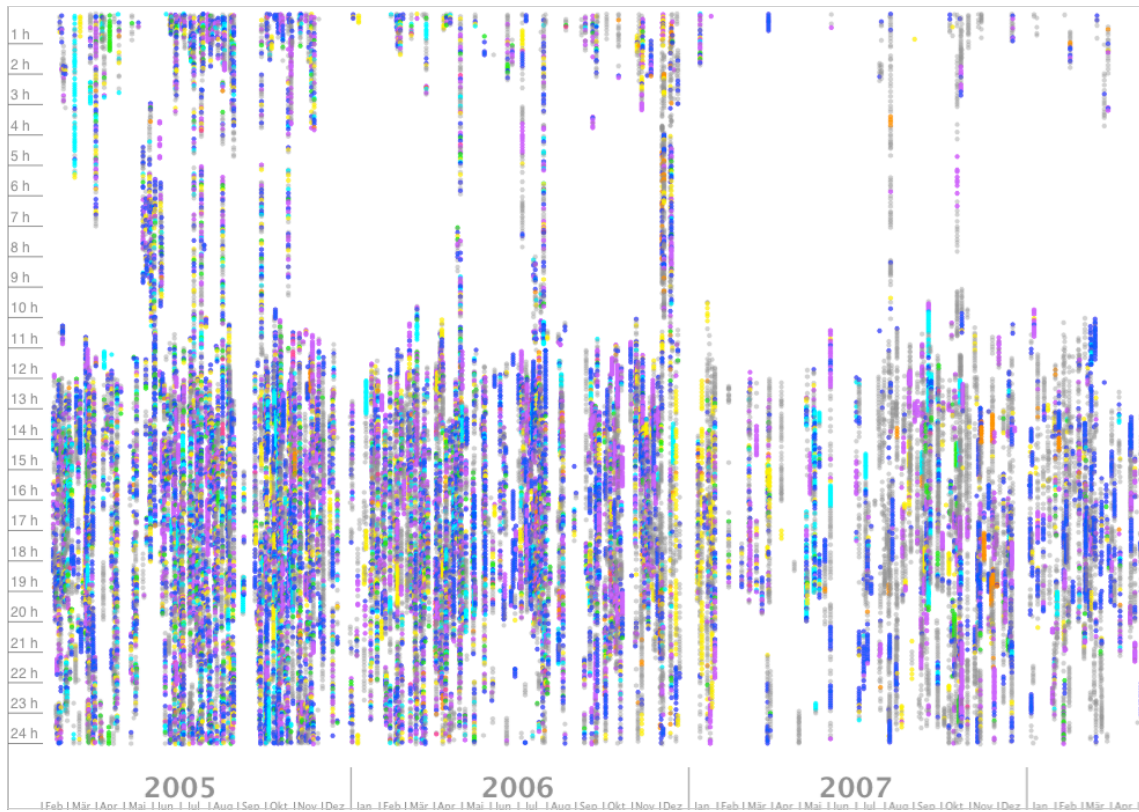


Figure 4.1. Music listening history of a user from the United Kingdom

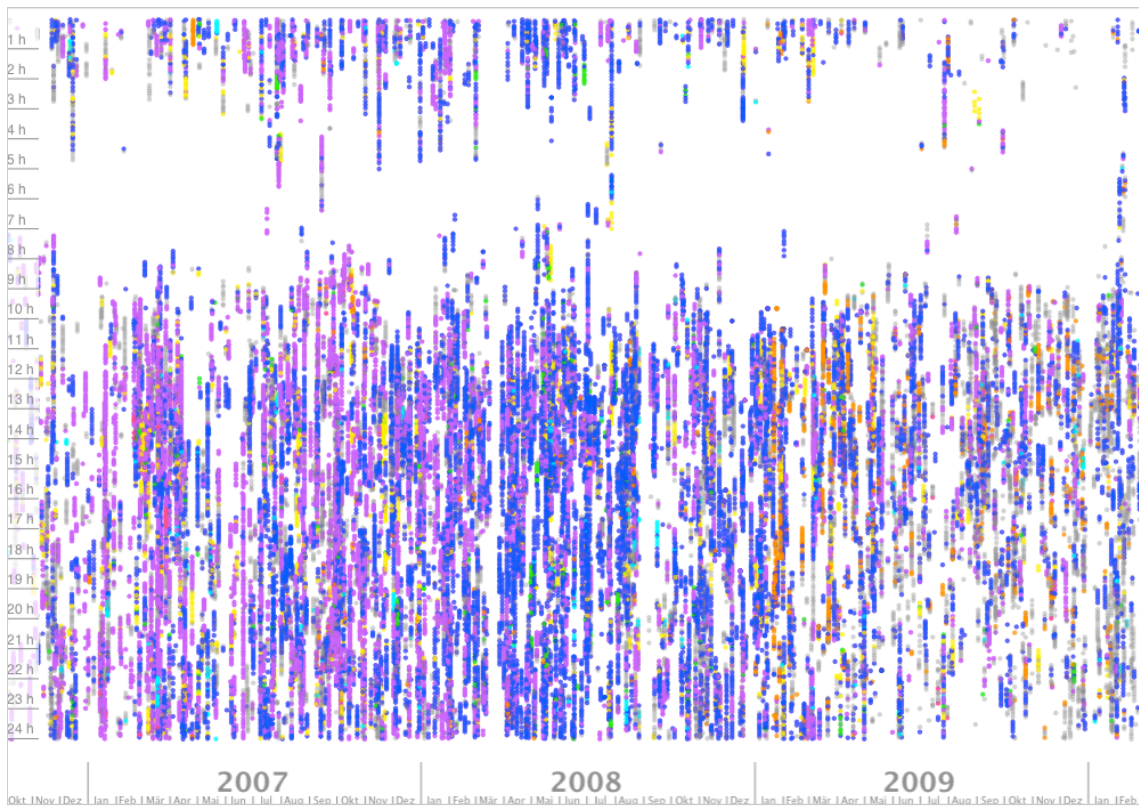


Figure 4.2. Music listening history of a user from Germany

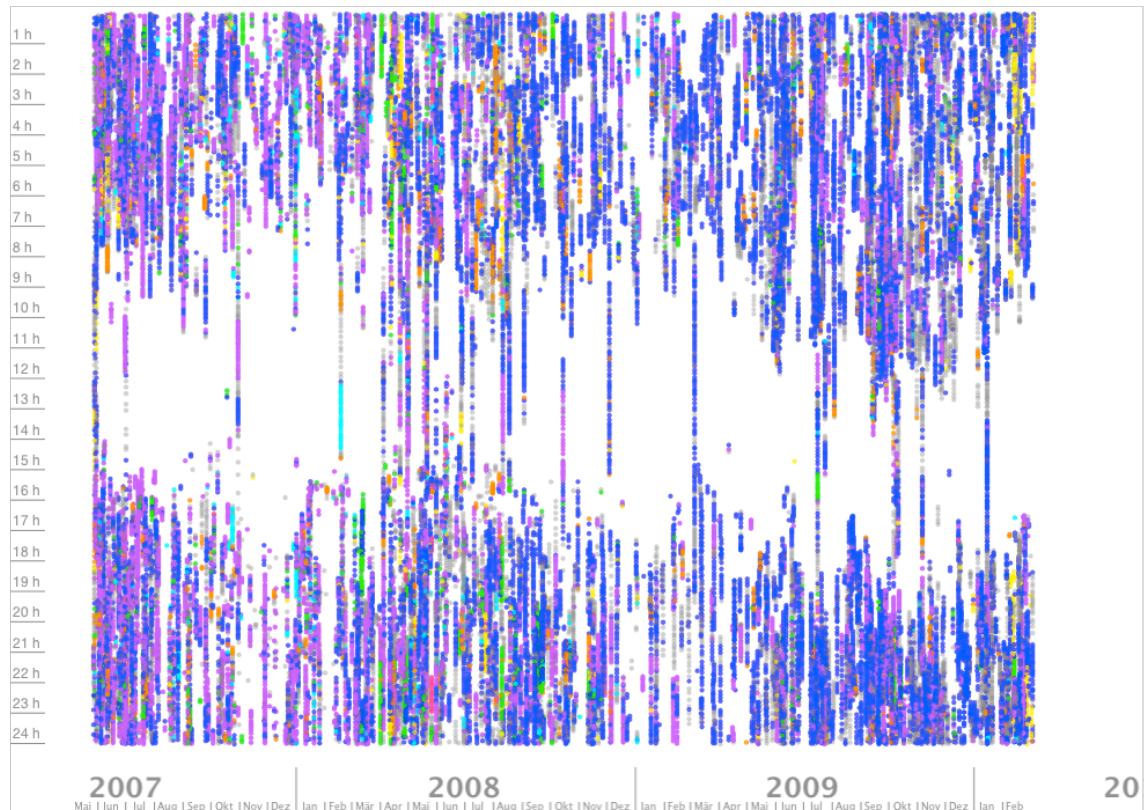


Figure 4.3. Music listening history of a user from the United States

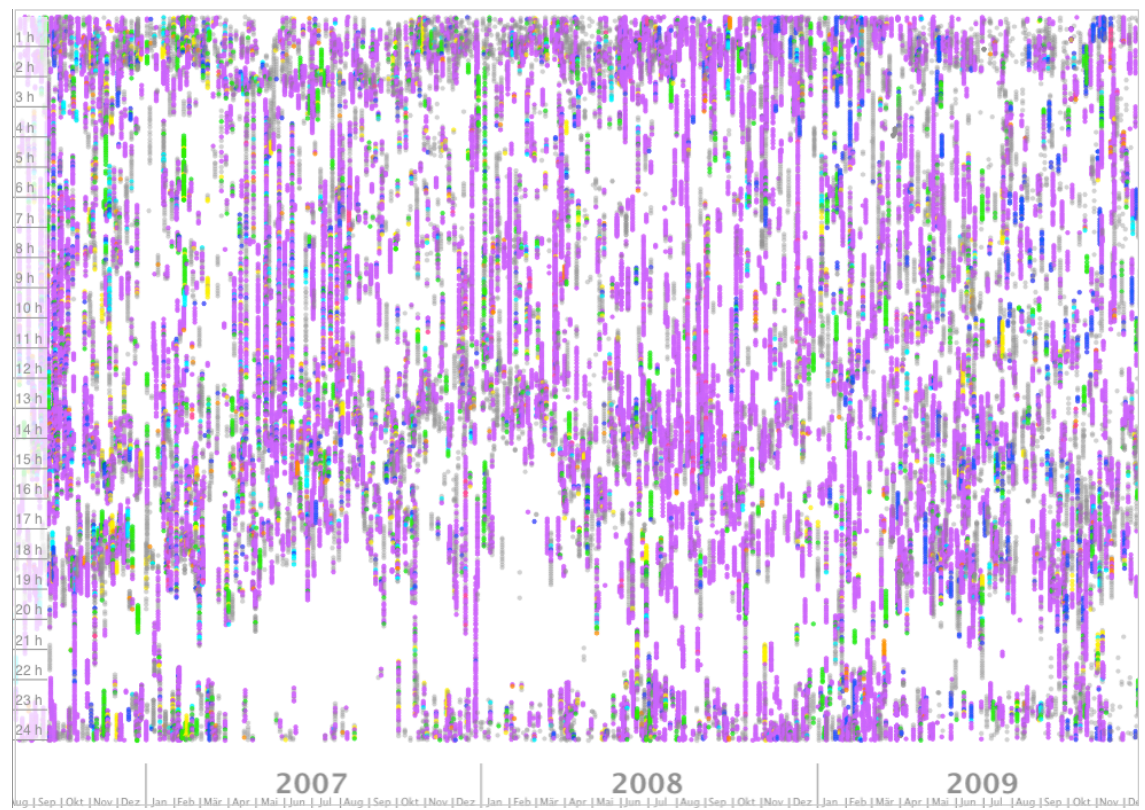


Figure 4.4. Music listening history of a user from Japan



Figure 4.5. Music listening history of a user from Austria

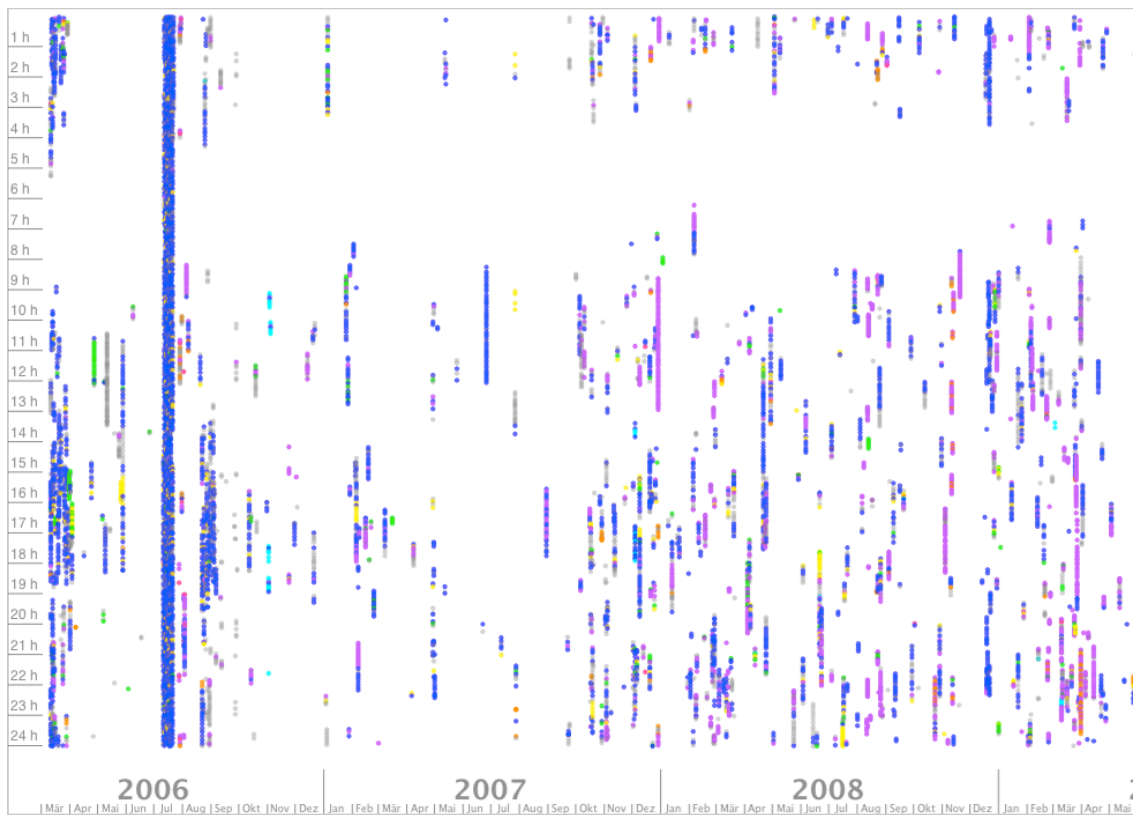


Figure 4.6. Music listening history of a user from Russia

5. Application Design and Implementation

The visualization was implemented as a native Mac application for Mac OS X v10.5 and higher in Objective-C using the Cocoa application environment. It takes advantage of a number of Mac OS X frameworks such as Core Data and Core Animation to achieve good processing and graphics performance on recent hardware even for large listening histories with more than 100.000 entries. Using these frameworks, it was possible to implement the entire application in just around 5000 lines of code (LoC).

An iterative approach was used to design and implement the application. Starting with a basic time-based visualization of a listening history, more functionality was designed, implemented and validated in each step. A considerable effort was done to keep the implementation modular and performant. This was achieved by refactoring the code whenever it became clear that a current design was not sufficient for demands posed by additional functionality, and by identifying and optimizing performance bottlenecks.

The following chapters give an overview of the general design of the program and show implementation details about important components, such as the data storage and visualization components, as well as the integration with other applications like iTunes, iPhoto and iCal.

5.1. General Design

The application is named „LastHistory“ and was designed as a document-based application using Cocoa’s *NSDocument* architecture in combination with Core Data for persistent storage. A LastHistory document constitutes an SQLite database managed by Core Data, which contains a listening history imported from a Last.fm user profile. A document is defined as follows:

„Conceptually, a document is a container for a body of information identified by a name under which it is stored in a disk file. In this sense, however, the document is not the same as the file but is an object in memory that owns and manages the document data. In the context of the Application Kit, a document is an instance of a custom NSDocument subclass that knows how to represent internally, in one or more formats, persistent data that is displayed in windows. A document can read that data from a file and write it to a file. It is also the first-responder target for many menu commands related to documents, such as Save, Revert, and Print. [...] When a window is closing, the Application Kit first asks the document, before it asks the window delegate, to approve the closing.“ [5]

LastHistory deviates from this definition in that it saves its documents automatically. This decision was motivated by the fact that LastHistory documents merely represent external data and are not edited in the traditional sense, where saving as well as undo and redo are common principles. Not asking the user to save the document thereby allows for a more seamless user experience.

A LastHistory document can be created in the application by supplying a Last.fm username for importing the listening history, as well as a file path for saving the file. Opening a document shows the visualization in a single window. Being a document-based application, LastHistory allows for handling multiple documents and therefore listening histories at the same time in separate windows.

Figure 5.1 shows a screenshot of a single document window. Each document window consists of a toolbar containing control elements and a search field at the top, the main view showing the visualization, and a status bar presenting status information and document statistics. The visualization consists of a timeline showing one or more „streams“ for the listening history and optionally other time-based user data like photos and calendar entries.

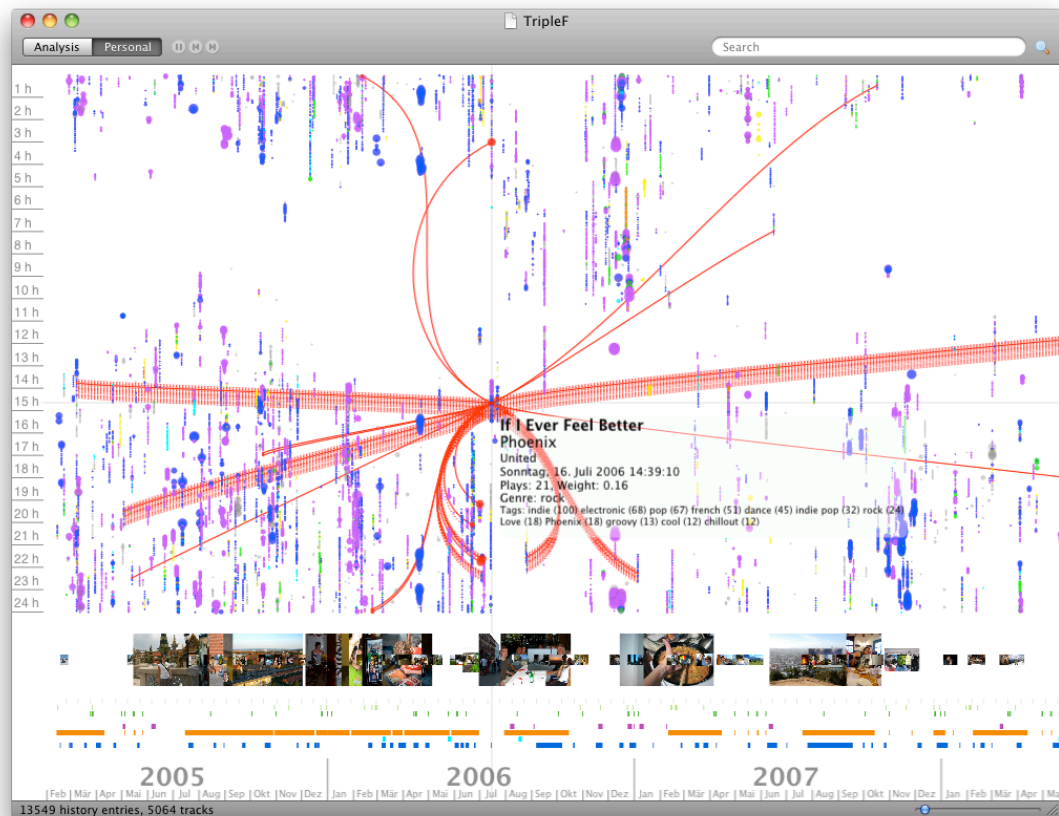


Figure 5.1. Screenshot of a document window in Personal Mode

5.2. Data Retrieval from Last.fm

Once a Last.fm username has been supplied, LastHistory will automatically download the listening history from the Last.fm profile with the specified username as well as information on the tracks in the listening history such as track tags and album using the Last.fm API [1]. The Last.fm API is a web service that offers access to the data collected and maintained by Last.fm, including user's listening histories, information about tracks such as album names, and user-generated content such as tags. The API responses consists of XML documents containing the requested information in a structured format.

Last.fm offers a number of API methods to retrieve various information on users, tracks, and artists. Following are the methods used by LastHistory:

- **user.getRecentTracks**

Given a Last.fm username, this method returns a list of recent tracks listened to by the user including timestamps specifying when the user listened to a track. The method returns up to 50 history entries at a time and has an optional „page“ parameter to retrieve older entries, thereby allowing to access the entire listening history of a user. For each track, Last.fm also supplies the name of the artist and album if available, as well as a unique MusicBrainz ID for the track.

- **track.getTopTags**

Given the name, artist, and optionally the MusicBrainz ID of a track, this method returns a list of social tags for the track including a normalized tag count between 1 and 100 specifying the relative importance of the tag. Social tags are free-text labels applied to the tracks by Last.fm users.

Data retrieval, of a user's listening history and associated tags, is implemented as Cocoa *NSOperation* objects, which encapsulate the task of retrieving the data from Last.fm and let the system run the task asynchronously on a separate thread if appropriate considering the available CPU cores and system load [2].

During retrieval, the data obtained through the Last.fm API is normalized and stored as a graph of Core Data objects in the document database. The following chapter describes the object model used for storing the data.

5.3. Data Model and Data Storage

The listening history data available from Last.fm is available as a list of entries consisting of information about the track, the artist, and the time the track was listened to. During development of the application, it soon became clear that this data needed to be normalized into separate entities in order to efficiently support the various queries the application was to perform on the dataset. Figure 5.2 shows an entity relationship diagram of the normalized data model. The colors of the entities indicate their relation. Entities belonging to the Last.fm user are shown in orange. Red marks entities constituting artists, albums, and tracks. Yellow marks entities for social tags.

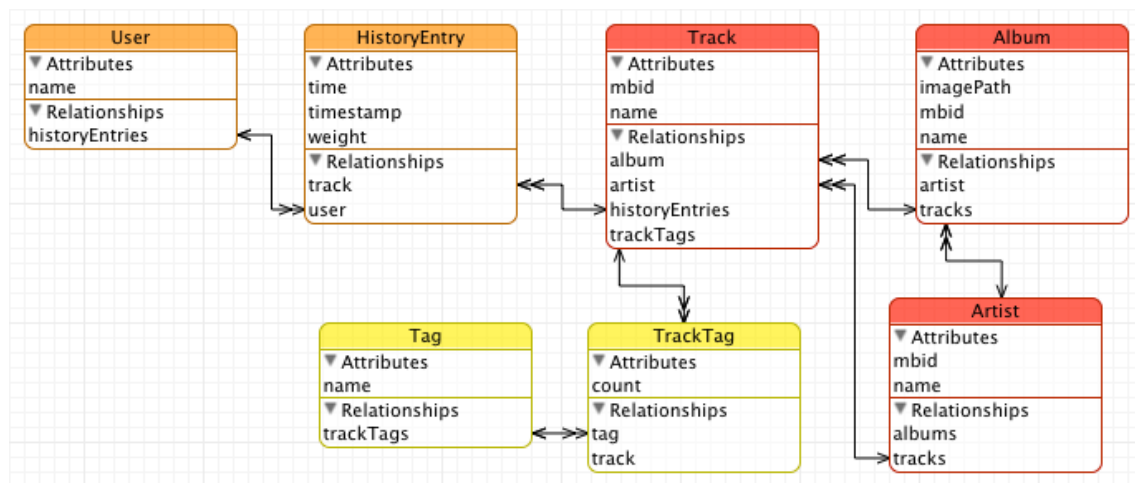


Figure 5.2. Entity relationship diagram of the data model

The Core Data data model framework [4] was used for creating and managing the data model and storing the data to disk in form of SQLite databases. Core Data allows the creation of the data model (called *managed object model*) in a graphical user interface as an abstract definition of the model objects (entities), complete with attributes and relationships between entities. At runtime, Core Data manages the object graph and mediates between the object graph and the underlying persistent store (the SQLite database) by allowing to create, fetch, and update managed data objects, without the need for the application to get involved in details about storage and retrieval.

Each LastHistory document manages its own Core Data persistence stack and associated database by using the *NSPersistentDocument* subclass of *NSDocument*. *NSPersistentDocument* pro-

vides a one-to-one mapping between the document and the backing object store. This is accomplished by providing a separate instance of *NSManagedObjectContext* for each document:

*„An instance of *NSManagedObjectContext* represents a single “object space” or scratch pad in an application. Its primary responsibility is to manage a collection of managed objects. These objects form a group of related model objects that represent an internally consistent view of one or more persistent stores.“* [6]

Creating and fetching managed data objects is done through methods provided by the object context of the current document.

5.4. Genre Classification

Since Last.fm does not supply a genre for a track, the genres are inferred from the track’s social tags from Last.fm. It has been shown that 68% of the tags created by users refer to genres [27], so using the tags for this purpose is obvious.

As the genres are used in the visualization for color-coding the nodes, only a limited number of genres could be used in order for the different genres to remain visually distinguishable. Seven genres were chosen to cover a majority of the genres listened to on Last.fm: classical, jazz, funk, hip-hop, electronic, rock, and metal. These genres were then assigned colors from the full color spectrum, except red which is used for highlighting playlists and active nodes in the visualization. Figure 5.3 shows a section of a listening history containing a variety of genres visualized through the different node colors.

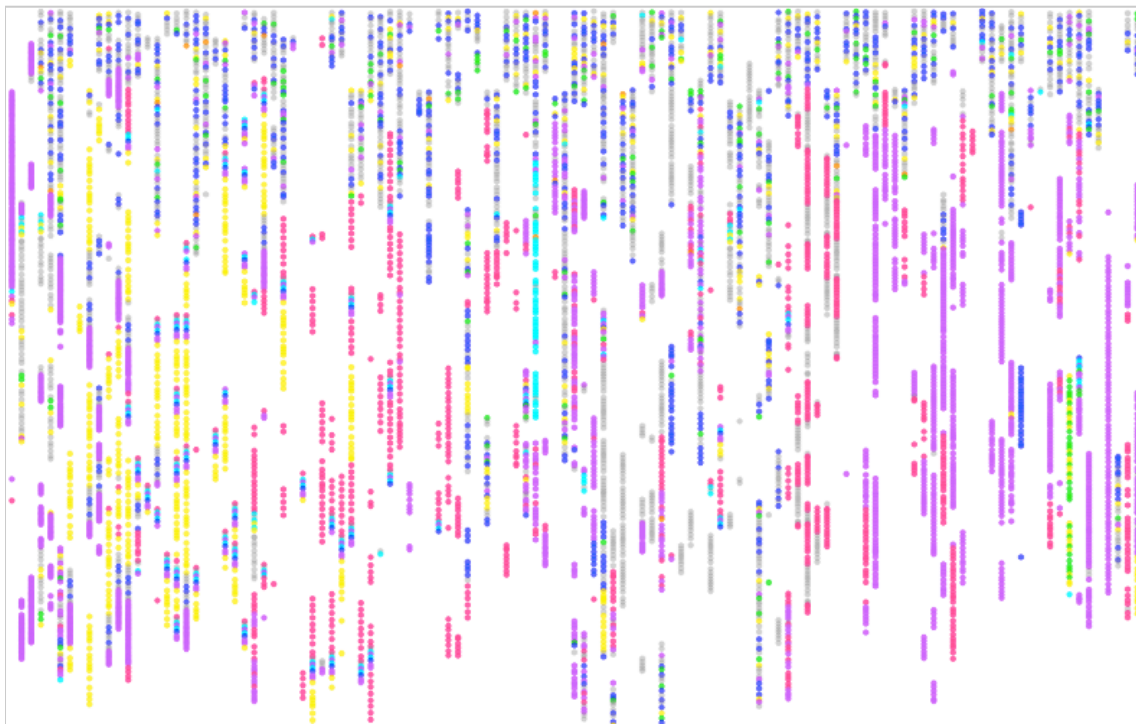


Figure 5.3. Section of the visualization of a listening history showing a variety of genres

To infer the genre from a track’s list of social tags, a list of tags was compiled for each genre, where each tag related to the genre itself or a similar one. For each track, the tag with the highest tag count appearing in any of the tag lists is used to determine the genre.

The mapping from tags to genres was created in an iterative process. By analyzing a random selection of listening histories, entries that had not yet been assigned a genre were identified, and one or more of the entry's tags were added to the list of tags of the genre best matching the entry. Following is the list of genres together with their tag lists.

- **classical**: classical, classic, piano, opera
- **jazz**: jazz, gospel, blues, swing, vocal jazz, jazz piano, new jazz, nu jazz, nujazz
- **funk**: funk, soul, disco, bossa nova, ska, reggae, roots reggae, world, worldmusic
- **hip-hop**: hip-hop, hip hop, hiphop, rap
- **electronic**: electronic, electronica, electro, electropop, house, minimal house, minimalistic house, electro house, electrohouse, electroclash, idm, trance, dance, club, techno, minimal techno, minimal, downtempo, down beat, downbeat, dub, chillout, trip-hop, drum n bass, dnb, drum'n'bass, drum and bass, drumandbass, jungle, dubstep, 2step, breakbeat
- **rock**: rock, classic rock, hard rock, indie, indie rock, punk, progressive rock, alternative, alternative rock, post-rock, post rock, piano rock, country, rockabilly, folk, new wave, synth pop, synthpop, singer-songwriter, singersongwriter, singer songwriter
- **metal**: metal, heavy metal, death metal, alternative metal

As can be seen from these tag lists, the mapping is in no way exhaustive and the genres are often a poor fit for the genre described by the tags. In some cases like the tag „folk“, the associated genre „rock“ certainly only applies to a subsection of the tracks tagged with „folk“. In other cases like „reggae“, the associated genre „funk“ might not be considered a super-genre by most people. As the primary focus of this work was not within the realm of genre classification, limited time was spent on creating the mapping and it was left for future work to improve it.

5.4.1. Visualization

LastHistory uses the Core Animation framework [7] for visualizing the music listening history and other streams. Core Animation provides a programming model to create visualizations and animations by composing an interface based on a hierarchical model of lightweight „layer“ objects. Layer objects provide the content to display and are arranged in a tree structure where each layer has a parent (the superlayer) and an optional collection of children (sublayers). For caching the layer content and compositing it to the screen as required, the layer tree is backed by a corresponding render tree. Interfaces can be animated by changing the properties of layer objects, whereupon Core Animation starts an implicit animation from the current to the new state by interpolating the values over the time of the transition. More complex animations can be created using keyframes, transformations, and explicit transitions. All animations are handled by the render tree on a separate thread, allowing the main thread to process further events while the animation is running. Core Animation can be hardware accelerated to offload some of the work of screen drawing onto the GPU in modern video cards. LastHistory uses animated transitions to make changes in the visualized data clear. This was shown to significantly improve graphical perception while interacting with statistical data graphics [28].

The applications layer tree consists of layers for interface elements, interaction, and actual visualization content. The root of the layer tree constitutes the base layer, which itself is without content but acts as a container for two sublayers: the layer containing the y-axis label, and a scroll layer that allows horizontal scrolling of the rest of the visualization (except the y-axis label). The scroll layer contains a layer for the x-axis label and one layer for each stream (the listening history stream, the photo stream, and the calendar stream), which are described in the following sections. Additional layers include layers for mouse-over information, the crosshair, and layers for highlighting the currently selected timeframe or event. All layers except the streams are managed by the *LHHistoryView* class, which itself makes up the view containing the Core Animation base layer. Each stream is managed by a subclass of *LHStreamLayer*. All layers

are instances of the Core Animation class *CALayer* or subclasses thereof. Figure 5.4. shows a screenshot of the application with markings for some of the layers.

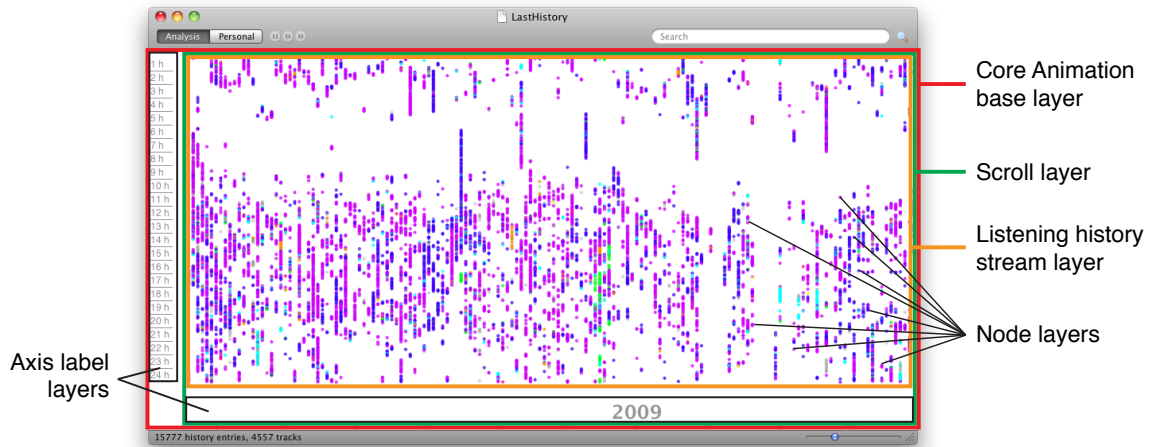


Figure 5.4. Core Animation layers making up the application interface

5.4.2. Listening History Stream

The listening history stream is managed by the *LHListeningHistoryStream* class. For each entry in the listening history, it creates a node layer and adds it as a child layer to the stream's base layer. The node layers are arranged horizontally and vertically inside the stream layer based on the history entry's date and time of day. Each node layer object holds a reference to the history entry object it represents and vice versa to facilitate showing mouse-over information, track sequence highlighting, and searching. Figure 5.5 shows an object diagram of the listening history stream and persistent history entry objects.

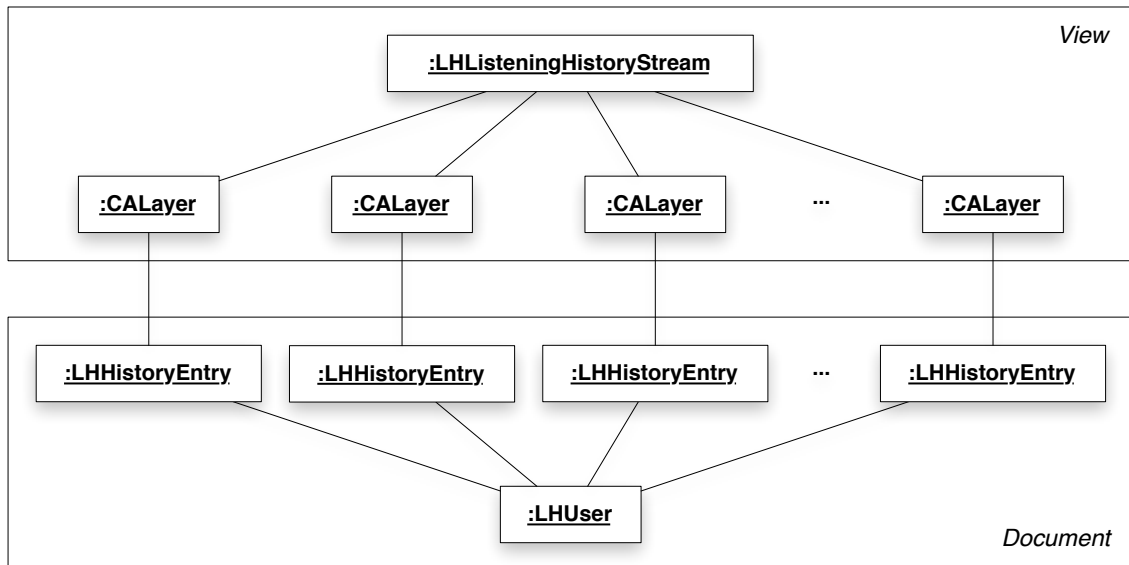


Figure 5.5. Object diagram of the listening history stream and persistent history entry objects

The content for each node is set to be an image containing a color-filled circle that is created at runtime according to a mapping table from genres to node colors. To reduce graphics memory,

the node image for each color is only created once for each color and then re-used for all nodes with the same genre. By default, all nodes are drawn at 75% opacity to increase visibility of overlapping nodes.

Node Scaling

To allow users to change the node size (e.g. to match different screen sizes), and for weighting nodes in Personal Mode, the node layers are scaled using a layer transform matrix without changing the content image. For this, the transform matrix T associated with each layer is changed as follows, where s is the scale factor applied to both the x- and y-axis:

$$T' = T \cdot \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Weighting of History Entries

By the process of weighting history entries, the application assigns significance to each node according to how intensely it was played in a 30 day period surrounding the entry in the listening history. This allows the user to identify tracks that bear a stronger meaning for that time period.

In the simplest form, the weight w_h for a history entry h of track $t(h)$ is calculated as

$$w_h = \frac{|t(h)^P|}{|t(h)|}$$

where

$$|t(h)| = \text{number of history entries for track } t(h)$$

$$|t(h)^P| = \text{number of history entries for track } t(h) \text{ in time period } P \text{ surrounding } h,$$

i.e. the percentage of history entries for the track that fall within the period P (here: $P = 30$ days). Given this formula, each history entry receives a weight in the range $[0,1]$, as both c_t and c_h are greater than zero and $c_t \geq c_h$.

The formula however disregards the amount a track has been listened to and would assign the same maximum weight of 1 to a track that was listened to just twice within 30 days in the whole listening history, and a track that was listened to 100 times total within the same time period. This leads to the improved weight w_h' to be calculated as

$$w_h' = \frac{|t(h)^P|}{|t(h)|} \cdot \frac{|t(h)|}{|t|_{\max}}$$

with

$$|t|_{\max} = \text{maximum number of history entries for any track in the listening history.}$$

Here the initial weight is multiplied with a factor indicating the amount the track was listened to in relation to the track that was listened to most in the listening history. This results in history entries of tracks that were listened to intensely throughout the listening history to receive a higher weight value than tracks that were only listened to a few times.

The last issue with this formula stems from the circumstance that a music listener's music library does not usually stay constant, meaning that people discover and start listening to new

music throughout their lives, while known tracks are sometimes listened to repeatedly throughout the listening history. This can result in tracks that were discovered early in the listening history to have a higher average number of history entries than tracks that were discovered later. In order to remedy this issue, the weight w_h'' introduces another factor m such that

$$w_h'' = \frac{|t(h)^P|}{|t(h)|} \cdot \frac{|t(h)|}{|t|_{\max}} \cdot m$$

with

$$m = \left(1 - \frac{M}{2}\right) + \left(\frac{d(h)}{d} \cdot M\right)$$

and

d = time interval between first and last entry in the listening history

$d(h)$ = time interval between first history entry and history entry h

M = constant describing influence of factor m on the weight (e.g. 0.5).

The factor m thus falls in the range $1 \pm M/2$, with the value 1 being assigned to a history entry that is exactly in the middle between the first and last entry of the listening history. The constant M is used to control the influence of m onto the final weight value. It was ascertained by analyzing various listening histories as to minimize the effect of a history entry's time point in the listening history on the weight value. A value of $M = 0.5$ was found to be optimal to this goal.

Using the final weight formula, tracks which only appear once in the listening history get assigned a weight that only depends on $|t|_{\max}$ and less so on $d(h)$. In order to get a fixed weight value for such tracks for all listening histories, history entries of tracks with only a single occurrence in the listening history get assigned a small default weight of 0.01, so that all such nodes appear with the same size in the listening history.

Finally, the calculated weight of a history entry is used in the visualization to scale the node size in Personal Mode such that history entries with higher weight values appear bigger than entries with smaller weights.

Highlighting of Identical Tracks and Track Sequences

When hovering over a node, the listening history stream will show appearances of the same track in the listening history connected through solid red lines. Other occurrences of the sequence of tracks the listening history entry is part of will be shown connected through dashed red lines. Figure 5.6. shows this schematically for two sequences of tracks.

The highlighting can be performed without any additional pre-processing by relying on the 1:n associations between unique tracks and history entries maintained by the normalized data model. Figure 5.7 shows the algorithm used for highlighting as pseudocode. The function *highlightHistoryEntry()* is called when the user hovers over a history entry node in the stream. It accesses the unique track associated with the active history entry and loops through all history entries for the track (i.e. all occurrences of track in the listening history), drawing a line from the highlighted history entry to each similar history entry.

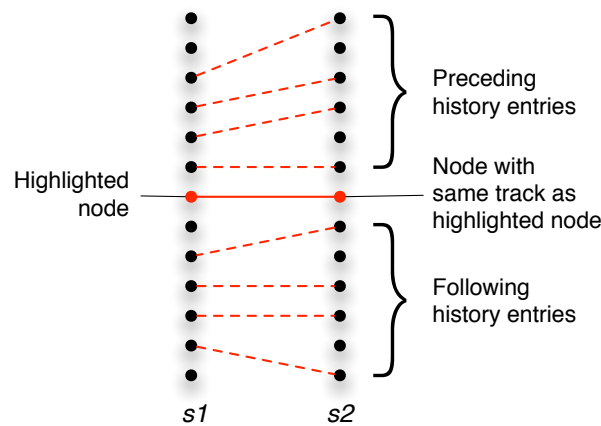


Figure 5.6. Schematic of detected identical tracks and track sequences

Detecting track sequences shared by the highlighted history entry and another history entry for the same track is split in two parts: the function *connectTrackSequences()* is called once for the list of history entries following the two entries in the chronological listening history, and once for history entries preceding the entries. Given the two sequences of history entries, the function loops through one of them and for each history entry checks the other sequence for a history entry with the same unique track. If two history entries with matching tracks are found, the algorithm then ensures a certain maximum distance *MAX_DIST* between their positions in the sequences and draws a dashed line between them. The maximum distance was set to 4 in the implementation, which means that two sequences can differ by at most three tracks. This allows sequences to still be detected if the user skips tracks in one listening session but not in the other.

In this simple implementation of the *connectTrackSequences()* function, given two sequences *s1* and *s2* each of length *n*, for each entry in *s1* the function in the worst case needs to loop through all of the entries in *s2* in order to find a matching entry (quadratic time complexity). In order to limit the run time of the algorithm for real-time use, the application's algorithm only looks at history entries that are no more than two hours from the highlighted history entry in the listening history, resulting in approximately 40 entries to be checked when assuming an average track length of 3 minutes.

```

function highlightHistoryEntry(historyEntry)

    // loop through other history entries for same track
    foreach similarEntry in historyEntry.track.historyEntries do
        if similarEntry == historyEntry then
            continue;

        // add solid line to similar entry
        addLine(historyEntry, similarEntry)

        // add lines between track sequences (forward/backward)
        connectTrackSequences(historyEntry.precedingEntries,
                               similarEntry.precedingEntries)
        connectTrackSequences(historyEntry.followingEntries,
                               similarEntry.followingEntries)

    end

end

function connectTrackSequences(sequence1, sequence2)

    foreach historyEntry in sequence1 do
        i1 = indexOf(historyEntry.track, sequence1.tracks)
        i2 = indexOf(historyEntry.track, sequence2.tracks)

        if i1 >= 0 and i2 >= 0 and ABS(i1 - i2) <= MAX_DIST then
            otherHistoryEntry = sequence2[i2]
            addDashedLine(historyEntry, otherHistoryEntry)
        end
    end

end

```

Figure 5.7. Pseudocode for highlighting identical tracks and track sequences

5.4.3. Photo Stream

The photo stream shown in Personal Mode is managed by the *LHPhotoStream* class and is positioned just below the listening history stream in the visualization. For each event in the user's iPhoto library that falls within the start and end date of the listening history, it adds a child layer showing the event's key photo to the stream's base layer. These layers are then positioned horizontally according to the event's start date. Hovering over a photo magnifies the layer and allows users to „scrub“ through all photos in the iPhoto event by moving the cursor horizontally within the bounds of the layer. Figure 5.8 shows a screenshot of the photo stream while the user is hovering over a photo event.

Since combining the listening history with the photos in the visualization mainly serves as a way to learn about connections between the two streams, photo events for which entries in the listening history can be found between the start and end date of the event are shown larger than events without corresponding listening history entries. This is again accomplished by applying a transform matrix to the layer.



Figure 5.8. Screenshot (section) of a photo stream

Clicking a photo event allows playing a slideshow of the event's photos along with the corresponding music from the listening history. Figure 5.9 shows a screenshot of the application showing a photo slideshow. The slideshow is implemented as a separate layer in the *LHEvent-ViewerLayer* class. By default, the application shows each photo for 3.5 seconds before fading to the next image. Controls at the bottom of the slideshow layer allow users to manually control the slideshow by pausing it and switching to the next or previous photo.

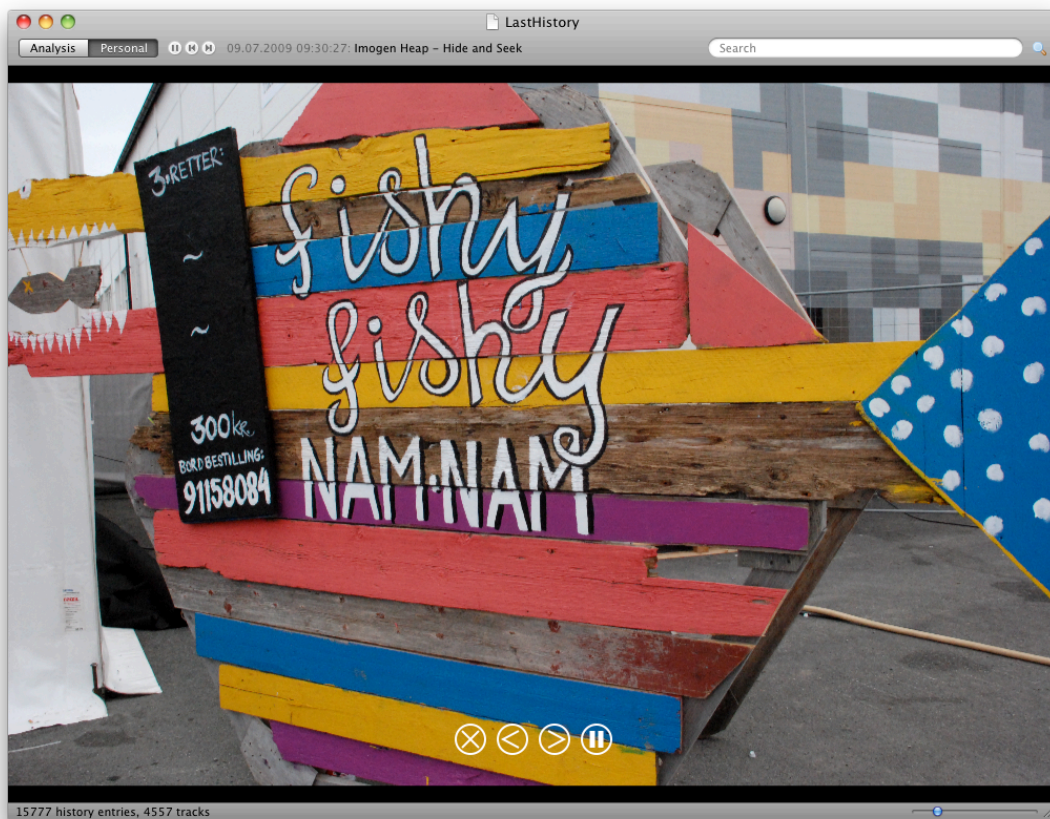


Figure 5.9. Screenshot of the application showing a photo slideshow

iPhoto Integration

While iPhoto does not provide an API to access its database, it maintains an XML file containing information about the user's albums, events, and photos including file paths. According to [9], this file (named „AlbumData.xml“) is „written out by iPhoto so that other applications, such as iMovie and iTunes, can read it and tell what photos and albums are in your photo library“. The file is never read back in by iPhoto, so that access to the photo library is read-only, which is sufficient for the purpose of displaying the photo stream.

While no official documentation on the XML format used by iPhoto exists, it was possible to decode the format sufficiently to access photo events and the associated photos. This was supported by fact that the XML file is written as a serialized „property list“:

„Property lists organize data into named values and lists of values using several object types. These types give you the means to produce data that is meaningfully structured, transportable, storable, and accessible, but still as efficient as possible. [...] The property-list programming interfaces for Cocoa and Core Foundation allow you to convert hierarchically structured combinations of these basic types of objects to and from standard XML. You can save the XML data to disk and later use it to reconstruct the original objects.“ [10]

The iPhoto property list contains object representations of the following types: dictionaries (collections of key-value pairs), arrays, strings, and numbers. Using dictionaries and arrays, the property list can be seen as a tree of objects. Figure 5.10 shows a diagram of the parts of the iPhoto property list accessed by the application. Rectangles represent keys in a dictionary, circles represent objects in an array, and text written in italic represents either strings or numbers. The „Master Image List“ key contains a dictionary of all photos in the library referenced by a unique key (a number), and the „List of Rolls“ key contains an array of all events in the photo library. The „KeyPhotoKey“ and „KeyList“ objects reference photo entries in the master image list dictionary and specify the event's key photo and the list of all photos in the event respectively.

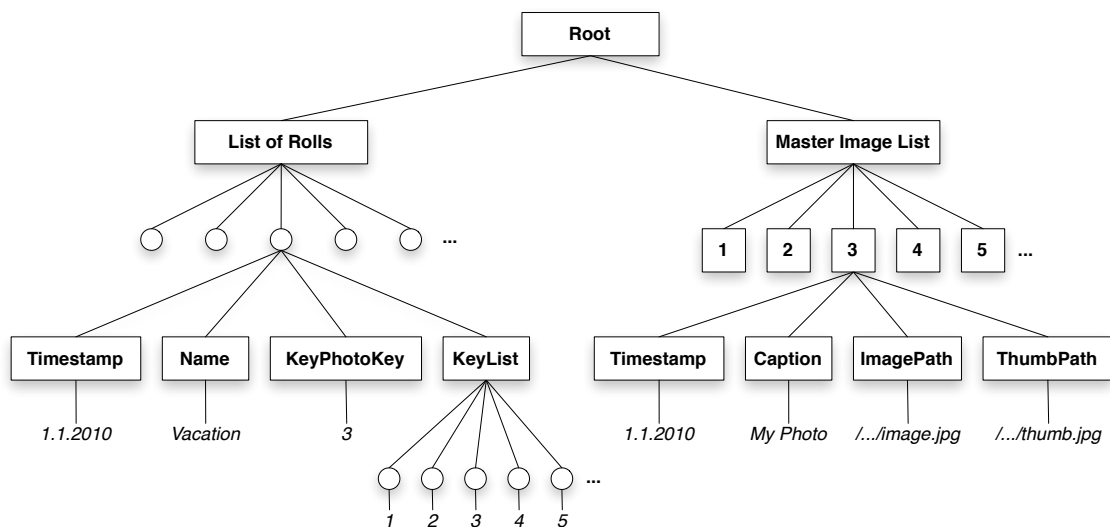


Figure 5.10. Diagram of the iPhoto library property list

5.4.4. Calendar Stream

The calendar stream is shown below the photo stream in Personal Mode and is managed by the *LHCalendarStream* class. When instantiating the stream, it first fetches every all-day events that falls within the start and end date of the listening history from the user's iCal calendar database using the Calendar Store API [8], resulting in a list of *CalEvent* objects. Each *CalEvent* object specifies a start date and end date for the event and is associated with one of the user's calendars. Each event is then displayed as a separate child layer of the stream's base layer.

Each of the user's calendars is assigned a separate vertical space in the stream for displaying all of its events. The event layers are displayed as rectangles with a fixed height within the space corresponding to the event's calendar. They are sized horizontally according to the number of days the event spans, and positioned according to the event's start and end dates. The event rectangles are filled with the color of their associated calendar, which can be set by the user in iCal. Figure 5.11 shows a screenshot of a calendar stream.

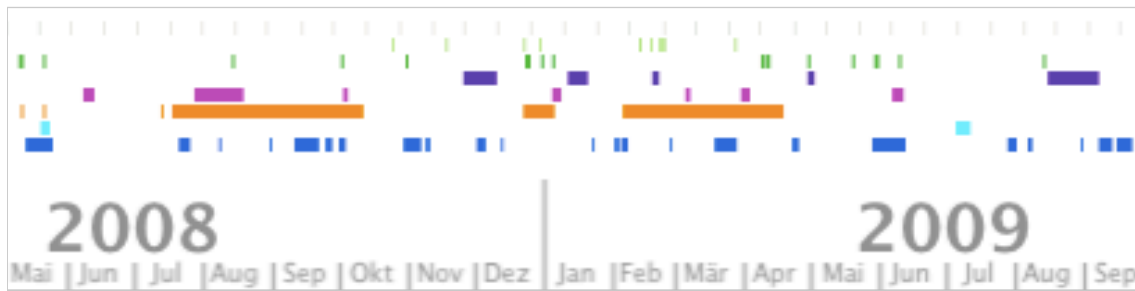


Figure 5.11. Screenshot (section) of a calendar stream containing events from 8 separate calendars

5.5. iTunes Integration

iTunes is a free media player application from Apple, which allows users to organize and play their digital music. LastHistory integrates with iTunes in two ways: it allows playing tracks from the listening history by accessing the user's iTunes library, and it allows saving track lists back to iTunes as playlists.

Playing Tracks from the User's iTunes Library

Similar to iPhoto, iTunes does not provide an API to access the user's iTunes library, but also writes out an XML file for read-only access by other applications. The file is again written in form of a property list (see „iPhoto Integration“ in Chapter 5.4.3), and contains a „Tracks“ dictionary with an entry for every track in the library. Figure 5.12 shows a diagram of some parts of the iTunes property list.

LastHistory uses the iTunes library XML file to find tracks on the user's hard disk when clicking a history entry node in the visualization. For this, the application performs a case-insensitive search of the name and artist of the track in the iTunes library using a hash table that is built when initially loading the iTunes library. Using the hash table, every track in the iTunes property list can be accessed with a key string of the form „artist - name“, thereby allowing efficient lookup of tracks in the iTunes library. Once a track has been found, the application uses the *NSSound* class that is part of the Cocoa API to play back the track's audio file.

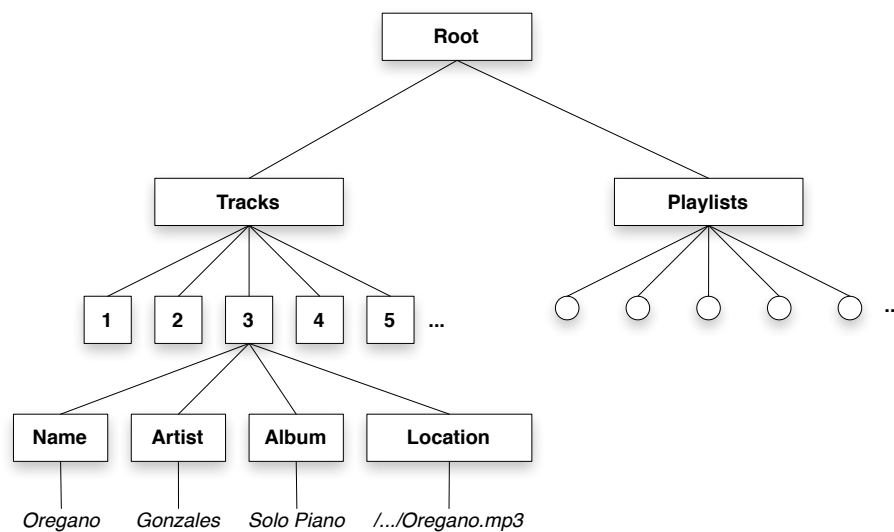


Figure 5.12. Diagram of the iTunes library property list

Creating Playlists in iTunes

When the „Charts Mode“ is enabled (e.g. in Personal Mode), the application allows playing tracks from a selected time period in descending order of the number of occurrences of the track in the time period. This allows users for example to quickly listen to the „top tracks“ from a certain year. LastHistory additionally allows to save the resulting track list as a playlist back to iTunes. This feature was implemented using the „Scripting Bridge“ technology that is part of Mac OS X, which allows to communicate with and control other applications:

„[...] A scriptable application is one that you can communicate with from a script or another application, enabling you to control the application and exchange data with it. To be scriptable, an application must define an interface for responding to commands. This interface and its implementation must conform to an object model (prescribed by Open Systems Architecture, or OSA) that specifies the classes of scripting objects, the accessible properties of those objects, the inheritance and containment relationships of scripting objects, and the commands that the scriptable application responds to. OSA packages commands to scriptable applications as Apple events and uses the Apple Event Manager to dispatch those events and receive data in return. [...] The Scripting Bridge framework makes it possible to send and receive Apple events using Objective-C messages instead of AppleScript commands or Apple-event descriptors.“ [11]

To create a playlist, the application first obtains references to scripting objects representing each track that should be added to the playlist by querying iTunes’ „master playlist“ using the „Persistent ID“ that is part of the iTunes property list entry for a track. These scripting objects are then added to a newly created „playlist“ object, that is in turn added to the list of user playlists, thereby creating the playlist in the iTunes library.

6. Evaluation

Two studies were conducted to evaluate the usability of the final application and to gain knowledge about the kind of insight users are able to generate using the application. The first was a small informal qualitative user study of participants using the application with their own listening histories. Additionally, an online survey was conducted by making the application available publicly on the Internet and asking users to fill out the survey after using the application.

6.1. User Study

The user study was conducted with four participants who had been recruited from university staff and students. The participants had to be Last.fm users with a listening history containing at least 5000 entries. Additionally, due to the program's architecture, they had to provide their own computer containing their digital photo library in iPhoto, and optionally calendar entries in iCal. The computers had to meet the minimum system requirements for the program.

The participants were between 21 and 27 years old, three male and one female. They had all been using Last.fm for 3,5 years or longer, and the size of their listening histories varied between around 23,000 and 78,000 entries. Two users were using the iCal calendar. Each session lasted around one hour, and was videotaped for later analysis. The „think aloud“ protocol was used to document the findings [29]. After a brief introduction to the program, the participants were asked to complete the following tasks:

- **In Analysis Mode:**

- Explore the user interface and interpret the visualization (nodes, node colors, connections).
- Inspect the visualization of the listening history to find insights in either the time, track, or genre dimension.
- Use the search feature to search for favorite genres and artists, and interpret the results.

- **In Personal Mode:**

- Interpret the different node sizes.
- Play photo slideshows for one or more of the photo events and report if the music played goes along with the photos.
- Select time periods with large nodes in the x-axis and report if the music played reminds you of the selected timeframe.
- Select different times of the day in the y-axis and report if the played songs correspond to songs predominantly played during the selected time of day.

After the sessions, the videos were analyzed and all findings by the participants were categorized into the three dimensions if possible (see [Chapter 4.1](#)). To give an overview of the results, figure 6.1 shows the total number of insights the participants had in each dimension.

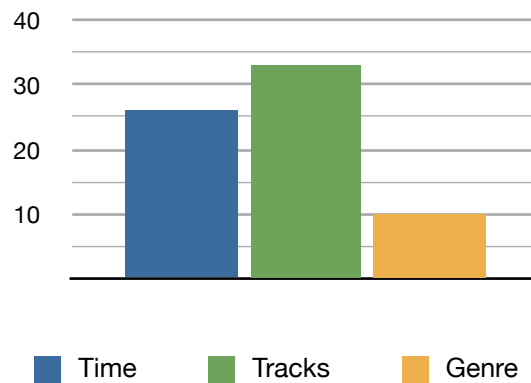


Figure 6.1. Total number of study insights per dimension

6.1.1. Study Insights in the Time Dimension

Among the first insight that all participants noted was the visibility of the daily routine in the visualization. While for some of them the lack of a daily routine was apparent, one participant noted how she gets up in the morning and first turns on music, before leaving home some time later. This was supported by a clearly visible band of nodes in the visualization between around 7:30 and 8:15. Another participant noted how he used to always listen to the same playlist in the morning before leaving the house, which the visualization again made evident by an hour-long single-colored band of nodes in the morning. Some participants observed how they rarely listened to music at night, while for others long lines of nodes at nighttime indicated long work nights, and one noted how „on most days I don’t start [the day] before noon“.

Changes in the daily routine were also apparent: one participant noted how „here I finished high school and then I didn’t have to get up as early in the morning“, pointing at a specific date in the visualization after which nodes became more sparse in the morning. Another participant showed where he worked for earning money to go abroad, getting up early in the morning and playing music as indicated by a dense section of nodes. Taking a job after finishing school was apparent in another case, where the participant showed how she mostly listens to music on the weekends after starting to work at a company. As an exception she found some nodes during the week, where she then remembered being on sick leave for a week.

Three participants showed specific times in the visualization where they remembered being on holiday, which was apparent by a lack of entries in the listening history. One participant pointed at an area with a clear shift in node distribution along the y-axis, explaining that he stayed abroad in a different timezone for half a year during that time. One participant explained a longer period of time with no history entries by switching to another computer and not installing the Last.fm client application. In another case a participant saw an overall increase in node density in the visualization in a specific year and explained how the Last.fm client application started submitting entries for tracks listened to on the iPod during that year, leading to more tracks being registered in the profile as the participant was using his iPod extensively.

6.1.2. Study Insights in the Tracks Dimension

All participants found different „phases“ in their listening history where they listened to a specific artist or track more often. One noted how he is a big fan of the Red Hot Chili Peppers, but using the application he found that he was listening to their tracks extensively two years ago, whereas in recent years he only had a limited number of their tracks that he still listened to. Another participant said he liked the Foo Fighters, and he found one period where he said that attending one of their concerts made him listen to their songs more, which was visible by an in-

creased density of nodes of tracks by the Foo Fighters. These findings were mostly made using the search feature by specifically searching for favorite artists.

Participants also found different „artist clusters“ in the listening history by browsing the visualization and pointing at nodes randomly. In several cases they traced the connection lines between occurrences of tracks to find that they had listened to a track multiple times within periods of several weeks, but not in between those periods. For example, one participant pointed at a song by Morcheeba, and found three clusters of nodes by this artist, each separated by about a year. These findings are in line with the observation by Andre et al. that people tend to listen to music in „waves“ [25]. In several cases, the first one of these clusters was explained to be coinciding with purchasing an album.

In other cases, participants connected artist clusters to specific events. When she discovered the cluster in the visualization, one participant noted how she tends to listen to one specific artist while studying for exams. Another found a long line of nodes where the same song had been played over a hundred times in a row, and explained that he had broken up with his girlfriend at that time. This case was especially visible in Personal Mode, where the song was highlighted with a large node size during that time in the listening history. In both cases, these clusters initially sparked interest and were investigated due to their node colors contrasting surrounding nodes, indicating that findings in the tracks dimension can be supported by varying genres in the listening history.

6.1.3. Study Insights in the Genre Dimension

Although the visualization of the track genres through node colors was interpreted correctly by all participants and often seemed to support sparking interest in exploring different sections of the visualization containing clusters of nodes with different colors, fewer actual insights were generated in this dimension than in the others. This might be due to the lower dimensionality of the genre dimension, with only seven genres in contrast to hundreds of days in the time dimension and several thousands of tracks the participants had listened to in the tracks dimension.

While one user did not express any insights in the genre dimension at all, and another only noted that the genres in the listening history were „pretty balanced“, the two remaining users made a number of comments referring to the node colors and genres of the tracks in their listening history. For these users, the change of colors in their visualization across time was especially apparent. One user noted how „2009 is color-wise more mixed than 2008“, referring to a larger amount of yellow (jazz) and orange (classical) nodes in 2009 that were not apparent in the year before. Another user realized that he had listened to mostly hip-hop in the first year after the listening history started, and had then gradually shifted to listening to more and more electronic music in the following years.

Grey nodes of tracks (representing unknown genres) were noted with a sort of pride by one user, noting that he often listened to what he called „underground music“ that had not been assigned social tags on Last.fm. Using the node colors, the participants were also able to find phases in their listening histories where they had listened to specific genres otherwise not as visible in the visualization. Two users referred to their „hardcore phases“ when they found clusters of pink (metal) nodes in the visualization.

6.1.4. Using the Personal Mode

The main activity in Personal Mode consisted of watching photo slideshows with the accompanying music, which all participants indicated they enjoyed. After using this feature, one participant explained that he would prefer using LastHistory over iPhoto itself for watching photo slideshows of his photo library.

Using the photo slideshow feature, the participants were able to recall in a number of cases listening to specific music for certain life events documented in their photo events. For example, one participant remembered listening to Cat Stevens as a way to relax between parts of a sports

tournament he was participating in. In another case a participant remembered listening to The Prodigy in the car while watching photos that had been taken during the journey. For another set of photos, the participant recalled attending a concert by Jason Mraz with a friend pictured in the photos. In all of these cases, the music in question was played by the application while watching the photos.

When playing tracks from a freely selected time period or time of day, the participants also indicated positive results regarding the choice of tracks by the application's „Charts Mode“ algorithm. When selecting the time between around 7:00 and 10:00 on the y-axis, one participant noted that the song played was part of a „wakeup playlist“ that he had been using for some time in the mornings. Similarly, when selecting the time between around 2:00 and 4:00 in the morning, the same participant said that the first song played he often listened to for going to sleep. When selecting a time period in the summer on the x-axis, another participant noted how The Kooks were a band that he primarily listened to in the summer when one of their songs was played for the selection. Another participant made a similar comment about a different song when again selecting a time period in the summer.

6.2. Online Survey

In addition to the user study, LastHistory was also made available as a free download online about four weeks prior to this writing, together with a 4-page introduction manual explaining the various application features. At the same time a short „teaser“ video showing different aspects of the visualization was published to spur interest by the online community.

The application was quickly picked up by a number of blogs from the infovis- [20], music- [21], and tech-scene [22,25] as well as the international press [19], and shared by hundreds of users on Twitter [23]. Within the four weeks, the application was downloaded over 3500 times, and the video was played over 8000 times.

Users who downloaded the application and whose computers met the minimum system requirements were able to simply enter their Last.fm username (or that of another person) in order to visualize their Last.fm profile. If a user would quit the application after more than 10 minutes, they were subsequently asked to fill out an online survey (see Appendix A). Alternatively, they could choose to fill out the survey from the application menu. In total 69 participants had completed the survey at the time of the evaluation, of which 66 stated to be male (95,5%) and 3 female (4,5%). The average age specified was 29,5 (minimum: 16, maximum: 67). What follows is a discussion of the remaining survey results, which were of both quantitative and qualitative nature.

6.2.1. Application Usage Results

One of the first questions asked the participants about their occupation, which was filled out 51 times. As shown in figure 6.2, a majority (43%) came from an academic environment including students, researchers, and professors. The second largest group (18%) stated an engineering-related job including software engineers and developers. Finally, 12% said to work as designers or in a similar profession. The remaining 28% stated other occupations ranging from „barrister“ to „office worker“.

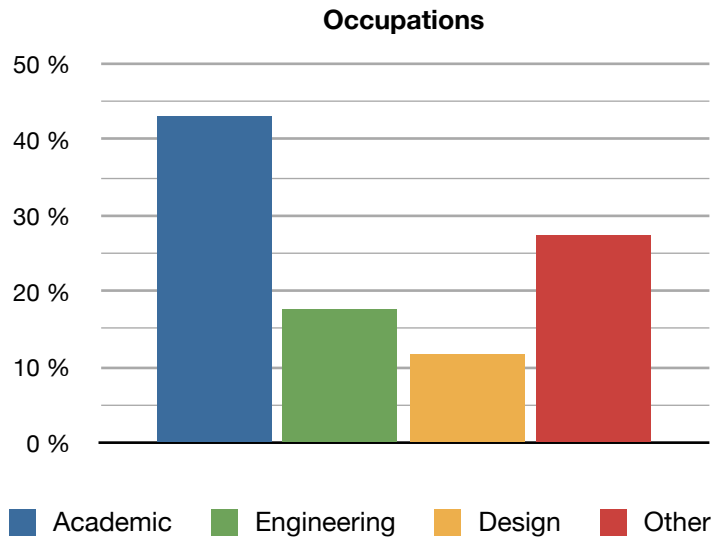


Figure 6.2. Online survey results: occupations

The next two questions were to find out about how the participants were using the application. First, users were asked about the different use cases (see figure 6.3): unsurprisingly, all participants said they used the application to analyze their own listening history, while 4% said they also used it for analyzing someone else's listening history. A large percentage (42%) stated to also use the application for reminiscing, and some for storytelling (6%). Answers for „Other“ (6%) included „interest in visualizations,“ „nice graphics and cool data mining,“ „discovering sleep and listening patterns,“ and „looking at the visualization techniques“.

The other question in this section asked users about which parts (streams) of the visualization they used (see figure 6.4). While nearly everyone used the Last.fm listening history (99%), 38% said they used the integration of photos, and 16% the calendar entries. Considering that LastHistory requires users to manage their photos and calendars in specific applications (iPhoto and iCal respectively), these numbers suggest that an integration with other photo and calendar services might open the Personal Mode to a wider audience.

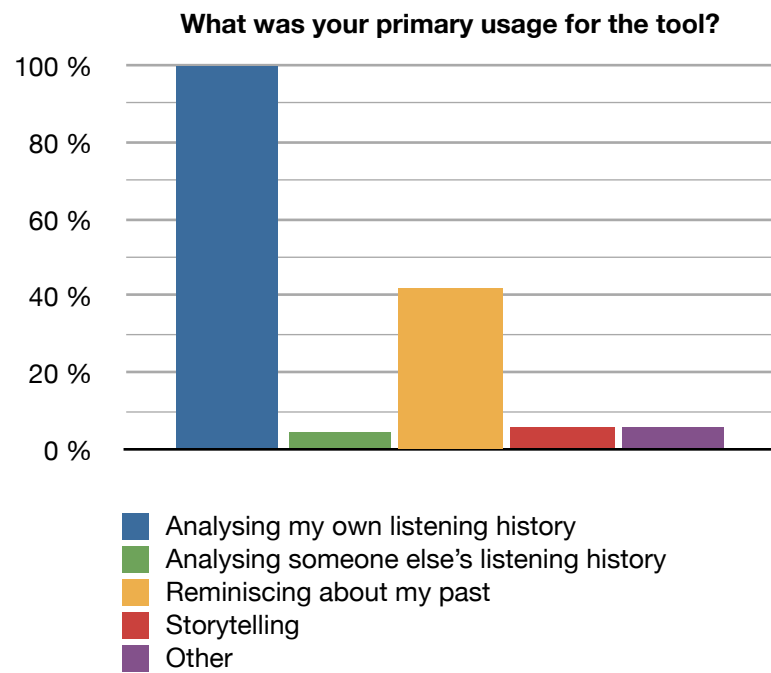


Figure 6.3. Online survey results: primary application usage

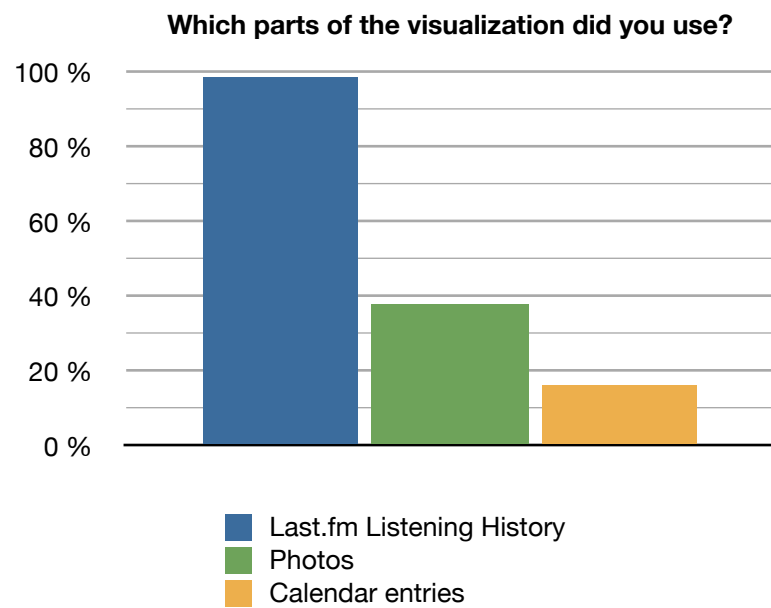


Figure 6.4. Online survey results: usage of different application aspects

6.2.2. Evaluation Results

The aim of the last section of the survey was to measure the usefulness of the application and the participants' satisfaction with the application usability. Based on the questionnaires by Lewis et al. [30], the users were asked to rate their agreement with each question on a 7-point scale, which was anchored at the end points with the terms "strongly agree" for 1 and "strongly disagree" for 7. Figures 6.4 and 6.5 show the median results for the usefulness and usability questions. The lower and upper „whiskers“ indicate the value distribution by means of the lower quartile and upper quartile respectively.

Overall, participants seemed to find the application very useful by agreeing that it helped them find interesting facts about a Last.fm listening history with a median rating of 2 (question A). They agreed slightly less when asked if it helped them to reminisce about their past with a rating of 3 (question B), which fits with the lower usage of photos and calendar streams described previously. This was also reflected in the question about the usefulness of the Analysis Mode versus the Personal Mode, where the former got a median rating of 2 (question C) and the latter a 3 (question D).

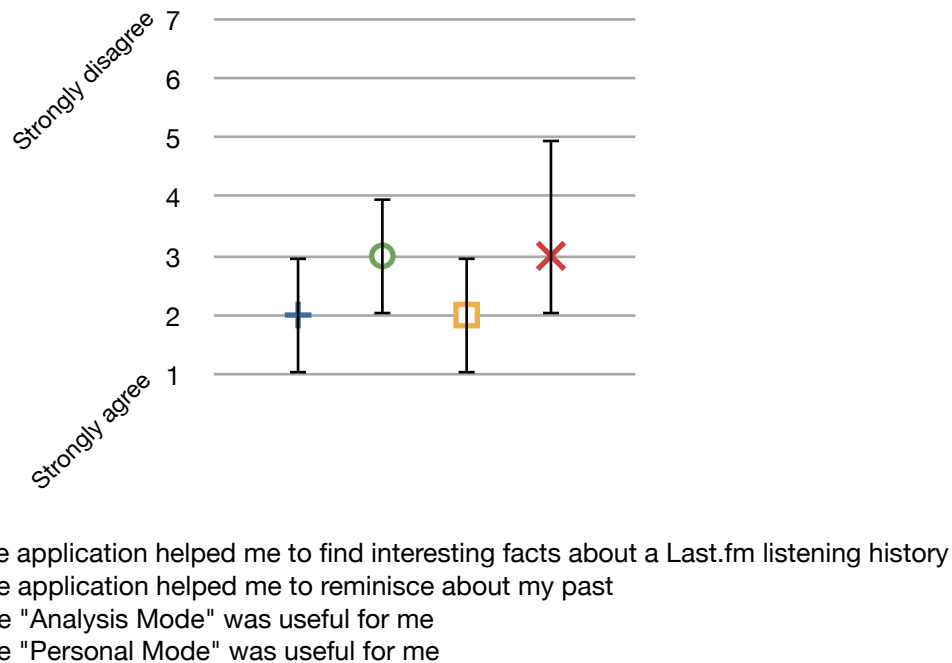


Figure 6.5. Online survey results: evaluation of application usefulness

The questions about the application usability were also answered very favorably (see figure 6.6). Participants gave a median rating of 2 both when asked if they were overall satisfied with how easy it was to use the application, and if it was easy to learn to use the application.

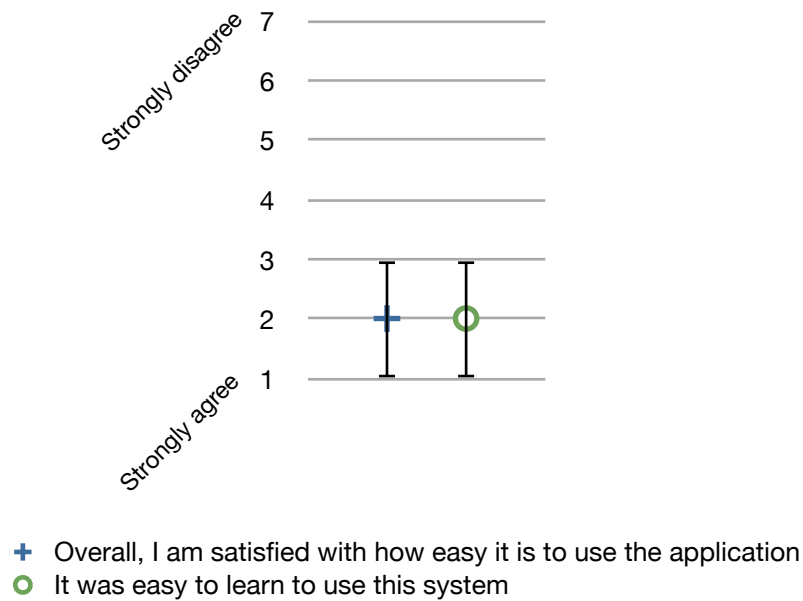


Figure 6.6. Online survey results: evaluation of application usability

6.2.3. User Feedback

Finally the questionnaire contained a number of free-form text fields where participants were asked to describe the insights they were able to generate using the application, and to leave feedback on how to improve the application. While not all participants filled out these fields, they contained some valuable feedback regarding usability and gave an idea about the type of findings users had about their listening histories.

Similar to the user study, insights were again categorized into three dimensions: time, tracks, and genre. As shown in figure 6.7, most insights reported by users were regarding time or tracks, while only a few participants reported insights in the genre dimension. This mostly fits with the results from the user study, although the number of insights reported in the time dimension was relatively higher in the questionnaire than in the user study.

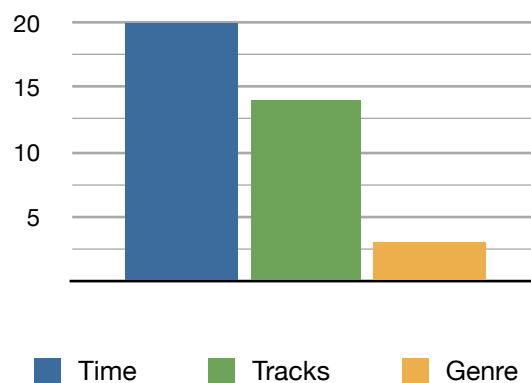


Figure 6.7. Total number of insights per dimension reported via the questionnaire

Looking closer at the participants feedback, following are some select quotations from the questionnaire.

- **Artifacts in the listening history:**

- *I sometimes fall asleep and leave my iPod on.*
- *Apparently I left iTunes running for 24 hours at one point.*
- *Leaving my iTunes running while I sleep.*
- *Huge gaps in my listening when I wasn't scrobbling.*
- *That I listened to music for 4 straight days. Apparently my computer was on and played the music without my knowledge.*

- **General patterns in music listening behavior:**

- *How all music comes and goes in waves.*
- *The times of the day that I tend to listen to music as well as periods of time in which I listened to one type of music or another.*
- *Pattern of when I listen to podcasts versus music.*
- *I go through waves of listening to genres.*
- *I stick to the same genre for a few weeks then move on.*
- *My Last.fm usage history is a pretty accurate way of determining those periods of my life when my insomnia was acting up.*
- *My listening hours.*
- *My listening time periods have changed, and I can identify those time periods with what's going on in my life.*
- *I was surprised to see that my listening pattern was pretty evenly spread throughout the 24-hour day, until the point where I got a 9-5 job, at which point it became almost entirely blank between 1am and 8am.*
- *I don't listen to much repeatedly in close succession, but over time I do.*
- *Periodicity of binges.*

- **Patterns regarding specific songs or artists:**

- *How often I listen to Grizzly Bear!*
- *That I listened to some songs much earlier than I thought.*
- *How I tend to listen to whole albums and not just individual songs.*
- *What music I listen to at nights.*
- *Which titles in my history I heard really often over a longer period of time.*
- *The lifecycle of various songs. Clicking on a random dot to see if it was at the end of the time I listened to it a lot, or during a time when I listened to it over and over again.*
- *Seeing exactly when I started listening to certain bands.*
- *How little I listened to some songs I thought very highly of.*
- *How several artists just pop up regularly, while others are just being heard during one period.*

- **Using the Personal Mode:**

- *Brought up memories.*
- *Lots of memories.*
- *Clicking on a photo gallery and listening to what I was listening to at the time was very powerful.*
- *The weighted display of important songs was mostly what I had expected, but there were a few songs in there that I had forgotten about, or didn't realize I had played so often.*
- *During the period where I wasn't often listening to much music during the night time, when I looked at the few instances of music playing in the early morning, I found that in many cases I could recall quite clearly the night that that happened, from the time and the tracks that were playing.*

A number of users also reported that they were not able to use the Personal Mode as intended, as they were not using iPhoto or did not have any photos on their computer, which explains the little feedback regarding Personal Mode. Some users also experienced crashes, which was found to be related to a threading issue in the initial version of the application that could occur when users started interacting with the visualization before all data was loaded from Last.fm.

Lastly, participants were asked how they would improve the application. Following are the main areas of improvement requested by users:

- Improving speed of importing the Last.fm listening history
- Improving application performance
- Improving navigation within the visualization
- Adding a legend to explain color coding of nodes
- Customization of color coding
- Enabling scrobbling when listening to songs using the application

7. Conclusions and Future Work

This thesis presented a full-featured application for visualizing music listening histories. The goal of this work has been twofold: to allow in-depth analysis of the data in different dimensions, and to support a secondary use case of making the listening history accessible to its creator for reminiscing and storytelling. The application was to be accessible for regular users, and able to handle real-world listening histories of considerable size.

The resulting visualization – named LastHistory – plots a music listening history on a two-dimensional timeline, and was shown to allow users identify patterns in a person’s music listening behavior in three basic dimensions: time, tracks, and genre. For example, in the time dimension the visualization allows learning about the amount of music listening, the regularity of the daily routine, the sleep-wake rhythm, and differences in listening behavior between weekdays and weekends; in the tracks dimension, possible insights are regarding the person’s listening patterns, such as the longevity and potential re-discovery of specific artists and tracks, and the listener type, for example if a person utilizes shuffle mode, or prefers listening to albums or manually-built playlists; and in the genre dimension, the person’s flexibility regarding different genres can be learned, as well as possible changes in taste from one genre to another.

To support the secondary use case, the visualization optionally includes photos and calendar entries. This personal data poses as landmarks to help make sense of discerned patterns in the context of a person’s life, and allows connecting events with the music listened to during these events. This has been shown to represent a powerful way to revisit past events and bring back associated memories. In this regard, the following comment from a user indicates the success of this approach:

„The killer feature for me is that you can listen to your full songs (since it uses your iTunes library, not a 30 seconds preview) and how you can relate your songs to your photos in iPhoto. The music that played with a slideshow for a certain trip was spot on and brought back more memories than the pictures alone.“ [20]

The visualization was implemented as a native Mac OS X application and takes advantage of frameworks for data persistence and hardware-accelerated graphics provided as part of the Mac OS X platform. This allowed supporting listening histories of considerable size, and the application was tested with histories containing over 150.000 entries.

Two user studies were conducted as part of the thesis to evaluate the application: an informal lab study and a large-scale online study. While the lab study provided valuable insights into the type of findings users were able to make about their listening histories using the visualization, the online study provided very positive results in the form of statistical data and general feedback. As part of an online user study, the application was made available on the Internet as a free download, and spurred considerable interest among Last.fm users and information visualization communities alike. Within just four weeks, the application was picked up by a number of international websites and downloaded over 3000 times, showing the general interest in the topic.

The feedback received from users also indicated a number of areas for future work. Improving the speed of importing the listening history from Last.fm was a common request, as downloading history entries and social tags took more than an hour for some larger listening histories. Currently, for each song in a history, the application has to make a separate request to the Last.fm web service for retrieving the social tags. A significant speed-up would likely require an improved API from Last.fm for allowing to download social tags for multiple songs at a time. Application performance was also noted as an issue for very large listening histories and slower computers, and remains a possible area for future work.

Users also noted two possible areas of improvement for the visualization itself. First, some users found it difficult to navigate in their listening history and lost the overview especially when using the slider to scale the timeline. A possible solution might be to add a separate view below the timeline that always provides an overview over the entire listening history, and highlights

the currently selected part of the visualization. A second area of improvement was regarding the node colors. One suggestion that was noted several times was adding a chart to explain the meaning of the different node colors, indicating that exploring the colors using the mouse-over information might not be sufficient. A few users also said their listening histories were colored primarily using one or two colors, as they were listening to a limited variety of genres. Adapting the color mapping to match sub-genres within a specific listening history might provide a way to increase the amount of information conveyed by the colors.

Finally, adding personal streams from sources other than iPhoto and iCal might provide additional insight for some users and open the personal use case to more users. Supporting other photo and calendar applications seems obvious, but other sources like blogs, news, or activities on social websites also come to mind. The Activity Streams protocol for syndicating user activities on social websites like Facebook and MySpace might allow accessing an interesting data set in this regard [26].

The application's source code was made available under a GPL license to the public [27], and it is the hope of the author to find other developers interested in improving the application and adapting it for other use cases.

8. References

- [1] O'Hara, K., M. Tuffield, and N. Shadbolt. Lifelogging: Privacy and empowerment with memories for life. In: *Identity in the Information Society*, 2008. **1**(1): p. 155-172.
- [2] Gemmell, J., G. Bell, and R. Lueder. MyLifeBits: a personal database for everything. In: *Communications of the ACM*, 2006. **49**(1): p. 88-95.
- [3] Bentley, F., C. Metcalf, and G. Harboe. Personal vs. commercial content: the similarities between consumer use of photos and music. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006: p. 667-676.
- [4] Van Dijck, J. Record and hold: popular music between personal and collective memory. In: *Critical Studies in Media Communication*, 2006. **23**(5): p. 357-374.
- [5] Card, S., J. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. 1999.
- [6] Pousman, Z., J. Stasko, and M. Mateas. Casual information visualization: Depictions of data in everyday life. In: *IEEE Transactions on Visualization and Computer Graphics*, 2007.
- [7] Ringel, M., E. Cutrell, and S. Dumais. Milestones in time: The value of landmarks in retrieving information from personal stores. In: *Proceedings of Interact 2003*, 2003: p. 184-191.
- [8] Plaisant, C., et al. LifeLines: visualizing personal histories. In: *Conference on Human Factors in Computing Systems*, 1996.
- [9] Carlis, J. and J. Konstan. Interactive visualization of serial periodic data. In: *Proceedings of the 11th annual ACM symposium on User interface software and technology*, 1998.
- [10] Van Wijk, J. and E. Van Selow. Cluster and calendar based visualization of time series data. In: *Proceedings of the 1999 IEEE Symposium on Information Visualization*, 1999: p. 4.
- [11] Baur, D. and A. Butz. Pulling strings from a tangle: visualizing a personal music listening history. In: 2009.
- [12] Chen, Y.-X., D. Baur, and A. Butz. Gaining Musical Insights: Visualizing Multiple Listening Histories. In: 2009: p. 1-5.
- [13] Havre, S., B. Hetzler, and L. Nowell. ThemeRiver: Visualizing theme changes over time. In: *IEEE Symposium on Information Visualization*, 2000.
- [14] Byron, L. and M. Wattenberg. Stacked Graphs - Geometry & Aesthetics. In: *IEEE Trans. on Visualization and Comp. Graphics*, 2008. **14**(6): p. 1245-1252.
- [15] Pretzlav, M. Last. fm Explorer: An Interactive Visualization of Hierarchical Time-Series Data. In: 2008.
- [16] North, A., D. Hargreaves, and J. Hargreaves. Uses of music in everyday life. In: *Music Perception*, 2004. **22**(1): p. 41-77.
- [17] Huynh, D., et al. Time quilt: scaling up zoomable photo browsers for large, unstructured photo collections. In: *CHI'05 extended abstracts on Human factors in computing systems*, 2005.
- [18] Bederson, B. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. In: *Proceedings of the 14th annual ACM symposium on User interface software and technology*, 2001.

- [19] Perlin, K. and D. Fox. Pad: An Alternative Approach to the Computer Interface. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, 1993: p. 57-64.
- [20] Graham, A., H. Garcia-Molina, and A. Paepcke. Time as essence for photo browsing through personal digital libraries. In: Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries - JCDL '02, 2002.
- [21] Goto, M. and T. Goto. Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces. In: Proceedings of the 6th International Conference on Music Information Retrieval, 2005.
- [22] Schafer, J., et al. Collaborative filtering recommender systems. In: Lecture Notes In Computer Science, 2007. **4321**: p. 291.
- [23] Aigner, W., et al. Visualizing time-oriented data - A systematic view. In: Computers & Graphics 31, 2007.
- [24] Aigner, W., et al. Towards a conceptual framework for visual analytics of time and time-oriented data. In: Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come, 2007: p. 721-729.
- [25] Andric, A. and G. Haus. Automatic playlist generation based on tracking user's listening habits. In: Multimedia Tools and Applications, 2006.
- [26] J. Levitin, D. This is your brain on music: the science of a human obsession. Plume. 2006.
- [27] Lamere, P. Social tagging and music information retrieval. In: Journal of new music research, 2008. **37**(2): p. 101-114.
- [28] Heer, J. and G. Robertson. Animated transitions in statistical data graphics. In: IEEE Transactions on Visualization and Computer Graphics, 2007. **13**(6): p. 1240-1247.
- [29] Ericsson, K. and H. Simon. Protocol analysis: Verbal reports as data (Rev. ed.). In: psycnet.apa.org, 1993.
- [30] Lewis, J. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. In: International Journal of Human-Computer Interaction, 1995. **7**(1): p. 57-78.

9. Web References

- [@1] Last.fm. Web Services. Jan 25, 2010. <http://www.last.fm/api/intro>
- [@2] Apple. Concurrency Programming Guide: Operation Queues. Jan 26, 2010. http://developer.apple.com/mac/library/documentation/General/Conceptual/ConcurrencyProgrammingGuide/OperationObjects/OperationObjects.html#//apple_ref/doc/uid/TP40008091-CH101-SW1
- [@3] Apple. Apple Portables: Two finger trackpad scrolling. Feb 19, 2010. <http://support.apple.com/kb/HT3448>
- [@4] Apple. Developing with Core Data. Feb 23, 2010. <http://developer.apple.com/macosx/coredata.html>
- [@5] Apple. Document-Based Applications Overview: Document-Based Application Architecture. Feb 28, 2010. http://developer.apple.com/Mac/library/documentation/Cocoa/Conceptual/Documents/Concepts/OverviewDocArchitecture.html#//apple_ref/doc/uid/20000023
- [@6] Apple. NSManagedObjectContext Class Reference. Feb 28, 2010. http://developer.apple.com/Mac/library/documentation/Cocoa/Reference/CoreDataFramework/Classes/NSManagedObjectContext_Class/NSManagedObjectContext.html
- [@7] Apple. Animation Overview: Mac OS X Animation Technologies. Mar 1, 2010. http://developer.apple.com/mac/library/documentation/GraphicsImaging/Conceptual/Animation_Overview/MacOSXAnimTech/MacOSXAnimTech.html#//apple_ref/doc/uid/TP40004952-CH6-SW1
- [@8] Apple. Calendar Store Programming Guide: Calendar Store Overview. Mar 2, 2010. http://developer.apple.com/Mac/library/documentation/AppleApplications/Conceptual/CalendarStoreProgGuide/Articles/CalendarStoreOverview.html#//apple_ref/doc/uid/TP40004770
- [@9] Fat Cat Software. The AlbumData.xml file. Mar 2, 2010. <http://www.fatcatsoftware.com/iplm/Documentation/iPLM/pgs/albumdata.html>
- [@10] Apple. Property List Programming Guide: Introduction to Property Lists. Mar 2, 2010. http://developer.apple.com/mac/library/documentation/Cocoa/Conceptual/PropertyLists/Introduction/Introduction.html#//apple_ref/doc/uid/10000048-CJBGDEGD
- [@11] Apple. Scripting Bridge Programming Guide for Cocoa: About Scripting Bridge. Mar 3, 2010. http://developer.apple.com/Mac/library/documentation/Cocoa/Conceptual/ScriptingBridgeConcepts/AboutScriptingBridge/AboutScriptingBridge.html#//apple_ref/doc/uid/TP40006104-CH3-SW9
- [@12] Last.fm. Scrobbling Timeline. Mar 7, 2010. <http://playground.last.fm/demo/timeline>
- [@13] Andrew Godwin. LastGraph. Mar 7, 2010. <http://lastgraph.aeracode.org/>
- [@14] Last.fm. Tube Tags. Mar 7, 2010. <http://playground.last.fm/demo/tagstube>
- [@15] Associative Trails. Normalizr. Mar 9, 2010. <http://www.normalizr.com/>
- [@16] Last.fm. Listening Trends. Mar 8, 2010. <http://playground.last.fm/demo/listeningtrends>
- [@17] Associative Trails. Normalizr Help. Mar 9, 2010. <http://www.normalizr.com/help/>
- [@18] Associative Trails. The atomic unit of attention. Mar 9, 2010. <http://www.associativetrails.com/blog/entry/The-atomic-unit-of-attention>

- [@19] Newsweek. LastHistory Mashs Up Your Music and Photo Timelines. Mar 10, 2010. <http://blog.newsweek.com/blogs/techtonicshifts/archive/2010/03/04/lasthistory-mashes-up-your-music-and-photo-timelines.aspx?print=true>
- [@20] FlowingData. Visualize your Last.fm listening patterns with LastHistory. Mar 10, 2010. <http://flowingdata.com/2010/03/03/visualize-your-last-fm-listening-patterns-with-lasthistory/>
- [@21] Music Machinery. LastHistory – Visualizing Last.fm Listening Histories. Mar 10, 2010. <http://musicmachinery.com/2010/02/16/lasthistory-visualizing-last-fm-listening-histories/>
- [@22] AppleWeblog. Visualiza tu historial musical con LastHistory. Mar 10, 2010. <http://appleweblog.com/2010/03/lasthistory>
- [@23] Twitter. „LastHistory“ Twitter Search. Mar 10, 2010. <http://search.twitter.com/search?q=LastHistory>
- [@24] Last.fm. FAQ - Scrobbling. Mar 14, 2010. <http://www.last.fm/help/faq?category=99#201>
- [@25] Lifehacker. LastHistory Graphically Visualizes your Last.fm History Through Time. Mar 17, 2010. <http://lifehacker.com/5493059/lasthistory-graphically-visualizes-your-lastfm-history-through-time>
- [@26] Diso Project. Activity Streams. Mar 17, 2010. <http://www.activitystrea.ms>
- [@27] GitHub. LastHistory source code. Mar 17, 2010. <http://github.com/triplef/LastHistory>

10. Appendix A: Online Survey

LastHistory Usage Survey

LastHistory is an interactive visualization for listening histories from Last.fm.

This questionnaire (which starts on the following page) gives you an opportunity to express your satisfaction with the usability of LastHistory. Your responses will help us understand what aspects of the system you are particularly concerned about and the aspects that satisfy you.

To as great a degree as possible, think about all the tasks that you have done with the system while you answer these questions.

Whenever it is appropriate, please write comments to explain your answers.

The survey will take 5-10 minutes to complete.
Thank you for your participation!

There are 13 questions in this survey

User questions

1 [10]Last.fm Username (optional)

Please write your answer here:

Please enter your username from Last.fm. This will help us evaluate your answers.

2 [20]Age *

Please write your answer here:

3 [30]Gender *

Please choose **only one** of the following:

- ☐ Female
☐ Male

4 [40]Occupation

Please write your answer here:

5 [50]What was your primary usage of the tool? *

Please choose **all** that apply:

- ☐ Analysing my own listening history
☐ Analysing someone else's listening history
☐ Reminiscing about my past

☐ Storytelling

☐ Other:

6 [60]Which parts of the visualization did you use? *

Please choose **all** that apply:

☐ Last.fm Listening History

☐ Photos

☐ Calendar entries

Evaluation

7 [20] Please indicate how strongly you agree or disagree with the following statements.

Please choose the appropriate response for each item:

	(strongly agree) 1	2	3	4	5	6	7 (strongly disagree)
The application helped me to reminisce about my past.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The application helped me to find interesting facts about a Last.fm listening history.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8 [30] What did you find that you were not aware of before?

Please write your answer here:

Please describe any findings that you had while using the application.

9 [40] Please indicate how strongly you agree or disagree with the following statements.

Please choose the appropriate response for each item:

	(strongly agree) 1	2	3	4	5	6	7 (strongly disagree)
The "Analysis Mode" was useful for me.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The "Personal Mode" was useful for me.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10 [60] What did you find out using the "Analysis Mode"?

Please write your answer here:

11 [70]What did you find out using the "Personal Mode"?

Please write your answer here:

12 [80]Please indicate how strongly you agree or disagree with the following statements.

Please choose the appropriate response for each item:

	(strongly agree) 1	2	3	4	5	6	7 (strongly disagree)
Overall, I am satisfied with how easy it is to use the application.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It was easy to learn to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

13 [100]How would you improve the application?

Please write your answer here:

Inhalt der beigelegten CD

Schriftliche Ausarbeitung

- Diplomarbeit als original Apple Pages '09-Datei und als PDF-Datei.
- PDFs der zitierten Web-Referenzen.

Vorträge

- Antrittsvortrag als PDF-Datei.
- Abschlussvortrag als PDF-Datei.

Implementierung

- Objective-C Quellcode von LastHistory
- Kompiliertes Programm
- Einführung zum Programm als original Apple Pages '09-Datei und als PDF-Datei

Video

- Teaser-Video das zusammen mit dem Programm veröffentlicht wurde