# MusicDigger: A Tool for Music Discovery

M. Adil Yalçın, Preeti Bhargava, Sravanthi Bondugula, Varun K. Nagaraja and Marcelo Velloso
Dept. of Computer Science
University of Maryland, College Park
yalcin@cs.umd.edu, prbharga@cs.umd.edu, sravb@cs.umd.edu, varun@cs.umd.edu, mvelloso@umd.edu

***Abstract*—Discovering music is a process that can be facilitated by many different approaches, depending on the music style and even personal preferences of the listener/researcher of music. The common ground for all approaches is to explore an artist's work as discographies and to detect collaborations between musicians. In this paper, we present MusicDigger as a tool that serves for these common discovery forms, based on a public-domain, detailed music metadata set consisting of millions of records. *Digging* is, at its core, discovering specific items within a database. With MusicDigger, exploration starts with an initial focus on an artist or record label. Our solution then can extend to two different interfaces. Our timeline based interface allows looking at artists (own or contributed) work with advanced filtering features and easy, intuitive browsing. Our network based interface, guided by the user, allows discovering collaborations between musicians. The nodes are placed in a circular layout and tree-like graphs are created, minimizing the number of edge crossings in music networks which are highly connected. Based on our user studies and interviews, we observe that our features are very helpful for people who regularly use a music database to learn about details and discover artists, and our interface provides a helpful and easy-to-navigate view on music data. We also believe that our interfaces and interaction options are applicable to other domains, such as browsing a database of movies.**

## I. Introduction

The world of music before the Internet was a place where musicians collaborated with each other locally and discovering new music outside of one's own country was difficult. Many talented individuals who were not as well-known often performed under the radar and as a result, their collaborations with other artists were not as well-advertised. The Internet has made it easier for musicians, especially lesser-known musicians, to connect to potential fans and collaborate with other musicians which has resulted in an explosion of data, cataloging the releases of artists and labels from around the world. With such vast amount of data available, it becomes necessary for tools to support easy exploration of the data, generate insights from it and help in discovering new music. MusicDigger is an application precisely designed to achieve an easy and intuitive way for digging this data.

We make use of a web site called Discogs[1] to access user-submitted data regarding the releases of musicians. Unlike other databases of music, such as Last.fm[2], Discogs strives for completeness of the information in the database and has set high standards for submissions and revisions of the database. Users can contribute to this site by entering details about a release that does not already exist in the database or modify an entry that currently exists. The users of the community then rate the accuracy of this data through a voting and comments system. The data is determined to be correct according to the masses through this system of community vigilance, much like web sites such as Wikipedia. MusicDigger uses the Discogs database of user-submitted data as its backend and adds functionality to visualize the information. We have two major visualizations to accomplish this: a timeline visualization to view the releases of an artist/label over time and a network visualization to view the connections of a particular artist/label to other artists/labels.

Our tool is neither a recommendation system like Last.fm, Pandora, nor a social music network explorer [1]. MusicDigger focuses on allowing users to discover music through the connections of artists and explore the music collection of artists in the database. Such an application is valuable for various kinds of users ranging from casual listeners, music lovers, and radio programmers to artists themselves. For example, we can see our application being used by radio programmers for selecting content that gets broadcast on a radio station as it aids in discovering new music easily through connections.

## II. Related Work

There has been extensive work done in the area of visualizing music data but much of it can be tailored to two main types of tools: music recommendation systems and tools for exploring personal music collections.

### *A. Music Recommendation Systems*

Music recommendation systems attract much attention due to the large market of users who are interested in listening to music that is similar to artists or genres they already listen to. Some of these systems are based on a community of users and their listening habits, such as Last.fm, and some are based on the acoustic properties of songs, such as Pandora[3].

Last.fm is a web site that uses a community-based approach to music recommendation. Users can submit the title and artist information of the songs they listen to through a personal account on the site and construct their own library of music to listen to. Using listening habits from the large number of users on the site, Last.fm can recommend new songs to users with similar tastes.

Pandora is a web site that uses the data in the Music Genome Project to make its recommendations. The Music Genome

[1]http://www.discogs.com/
[2]http://www.last.fm/
[3]http://www.pandora.com/

Project has isolated over 400 different musical attributes and it tags songs and artists with these attributes. The user is then given the option to create radio stations based on a particular song/artist. Pandora finds other artists/songs that are similar to a particular song/artist and plays its recommendations to the user like a music radio. This method of recommendation differs from Last.fm because the Music Genome Project has come up with musical attributes and classified songs/artists using experts in the field of music as opposed to a community-based approach where users are not necessarily music experts.

### *B. Tools for Exploring Personal Music Collections*

MuVis, MusicBox, Variations2 and Mambo are some of the software applications that allow a user to explore their personal music collections in different ways.

MuVis [3] focuses on visualizing a large collection of music using treemaps. The ordering of the treemap is based on a pivot artist selected by the user that gets displayed in the top-left corner. Other artists in the music collection are mapped according to their similarity to the pivot artist. The user is allowed to control other variables such as the size and color of each node in the treemap and the user can filter results by the duration of the track, the release year, the genre, the beat, and the mood.

MusicBox [4], on the other hand, maps a music collection onto 2D space by applying principal components analysis (PCA) to contextual and content-based features. After projecting the songs in a collection onto 2D space, the MusicBox application then plots a scatter diagram of the songs. The guiding intuition in navigating this visualization is that similar songs are closer together.

Variations2 [5] focuses on using a grid to display the search results of a personal music library and providing details about the selected entry from the results. When the user searches for a term that is common to a few artists, the search results show up with genre on the x-axis and artist on the y-axis. If the user selects a particular composition, the data is now displayed with year of composition on the y-axis instead of the artists.

The authors of Mambo [2] have designed a zoomable UI to browse through music collections on mobile devices. They call their widget the FacetZoom, an interactive tree visualization. Each level in the tree is displayed as a horizontal bar divided into cells that are equal to the number of nodes at that level. Only a subset of levels are displayed at any time to allow for fast traversal through the music list. The zoom widget is accompanied by a tabular data display which displays the data as arranged using the FacetZoom widget. When there is a large dataset, the authors suggest a space-filling layout algorithm that calculates the largest possible rectangle for each item so that all items fit into the available space.

Apart from these tools, there are several ideas that have been proposed to visualize music collections. Torrens *et al.* [6] suggest three different ways of graphically visualizing music libraries considering five criteria - genre, artist, year and playcount, rating, and added or last played date. They first propose a disc visualization to provide a good overview of percentages and proportions. The radius of the disc can be regarded as the time axis, the size of a sector is proportional to the number of tracks of the associated genre with respect to the whole library, and each sector is further split into sub-sectors representing the artists of the associated genre. Next they propose a rectangle visualization similar to the disc visualization that uses rectangles instead of discs. The time axis in this visualization goes along the vertical axis. Finally, they propose a treemap visualization where the size of the region corresponds to the number of tracks for a particular genre.

Adamcyzk [1] discusses the topic of effective information visualization for exploring musical social networks. The author proposes a network model based on relationships between music artists gathered from a site AllMusic[4]. The links between artists are generated by using the information in the similar artists section who are chosen by AllMusic's own experts and is influenced by user feedback. Metrics like centrality and betweenness are embedded in the network by mapping it to shape and size of the nodes respectively.

## III. Data Description

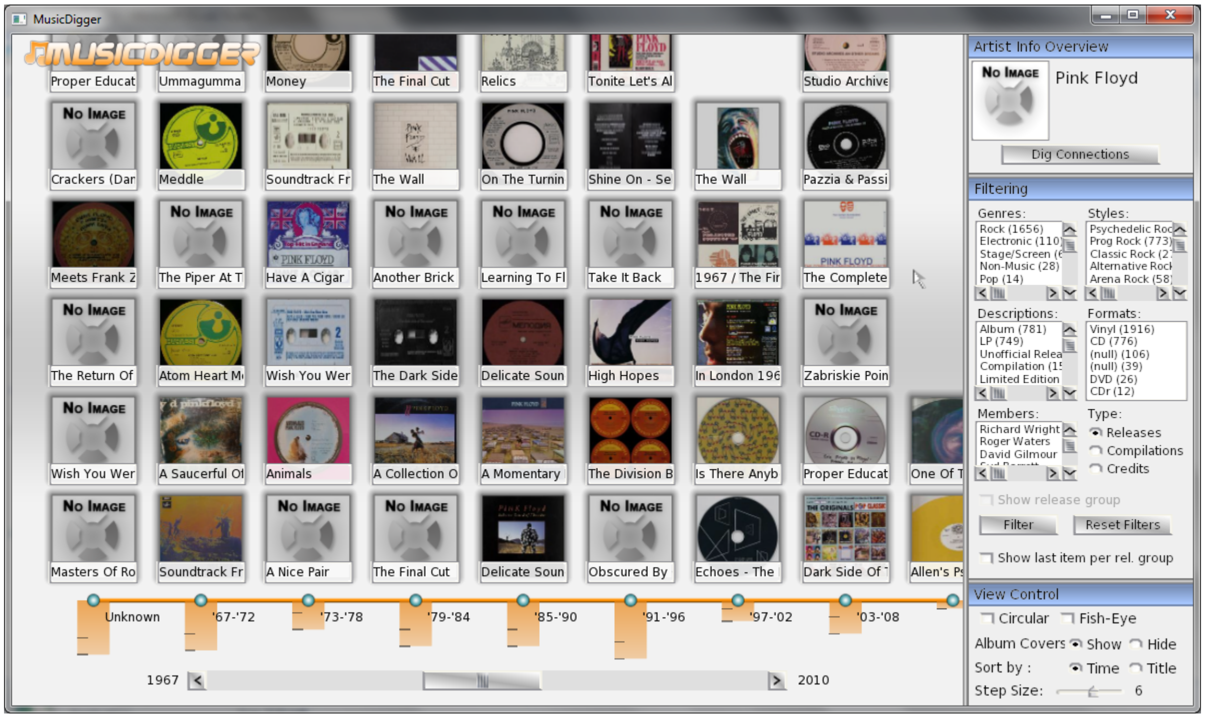
MusicDigger is based on the public-domain meta-data distributed by discogs.com. Discogs identifies the data under three main categories - artists, releases and record labels.

Artists can be individuals (Ex: Eric Clapton) or groups (Ex: Beatles, Pink Floyd), who have a set of releases referred to as their discographies. Releases are tangible items that the artist has produced officially, or even unofficially. Each release item can have a date of release and can be made of multiple items in different formats (Vinyl, CD, DVD, Cassette, electronic file, etc). The data includes additional information such as limited editions, and production details, such as mixed compilations, as well. Each track includes a detailed list of songs and credit/personnell information both for the release as a whole and for each track, if available. Discogs users also can attach one or more genre (rock, electonic, jazz, etc) and style (sub-genres) tags to a release.

Record labels (Ex: Sony Music, Blue Note) are the companies that deal with marketing releases, and they can be used to categorize music into styles and artist communities, and can convey information about the style of an artist or an album in many cases. Labels can have parent and sub-labels (ex: specializing in a style or associated with a specific country). Each release can link to one or more label as indicated in the sleeve of the actual item.

Discogs allows users to access monthly data dumps of the entire Discogs database[5]. The data includes all the releases users have contributed to the database when the dump was made and it is provided in the XML format specified by the Discogs API. The Discogs API allows users to remotely request the data for releases using the release id. We have

[4]http://www.allmusic.com
[5]http://www.discogs.com/data/



Type: [✓ Releases] Compilations Credits

Album Covers [✓ Show] Hide

Sort by : [✓ Time] Title

Step Size: 6

Fig. 1. Pink Floyd's Discography

made use of data from the latest database dump made in May 2011.

## IV. Visualizations

The two main visualizations in MusicDigger are the timeline visualization and the network visualization.

1) Timeline visualization: The releases of an artist or a label are displayed on a timeline making effective use of the display space, along with various other filtering and view controls.
2) Network visualization: Connections are identified between an artist of choice to other artists based on collaborations in releases, links through other artists/labels etc.

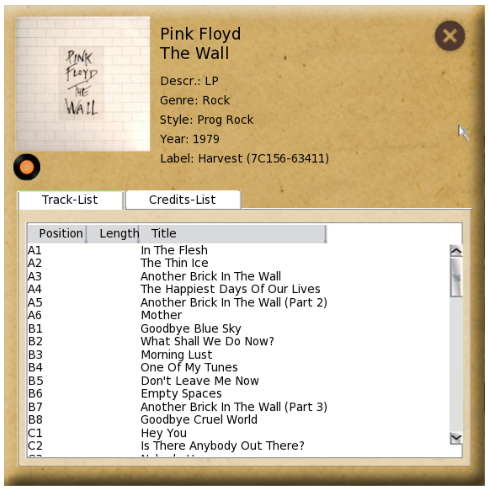
### A. Timeline Visualization

Timeline visualization, shown in Figure 1 allows the user to view the releases of artists or record labels in chronological order of release year or alphabetical order of release title. The main aspects of this visualization are:

*1) Alpha/Time-Dot:* Releases are grouped into bins (represented by dots) along the time-line based on release year or title. In the initial view, the size of the bins is automatically selected such that entire range of the releases is visible on the screen. All the releases belonging to a bin are then stacked up as album arts above the dot. Since the time-line is used to sort the releases by year or title, the range can be represented by years or letters. When the number of stacked releases becomes greater than the number that can be displayed vertically, the list can be scrolled up or down by single click and then dragging the mouse.

*2) Step Size:* A step size control is provided to the user to adjust the number of years or letters assigned to each bin. Increasing the step size decreases the total number of dots required on the time-line and hence reduces the horizontal space required to display the releases. Similarly decreasing the step size will provide a closer view to a particular year or letter while the entire time-line occupies a large horizontal space.

*3) Bar-Chart:* A bar chart below the Dots indicates the relative number of releases belonging to a Dot. Since the users can get lost while scrolling through the stacked releases, a pair



Pink Floyd
The Wall

Descr.: LP
Genre: Rock
Style: Prog Rock
Year: 1979
Label: Harvest (7C156-63411)

| Position | Length | Title |
|---|---|---|
| A1 | | In The Flesh |
| A2 | | The Thin Ice |
| A3 | | Another Brick In The Wall |
| A4 | | The Happiest Days Of Our Lives |
| A5 | | Another Brick In The Wall (Part 2) |
| A6 | | Mother |
| B1 | | Goodbye Blue Sky |
| B2 | | What Shall We Do Now? |
| B3 | | Morning Lust |
| B4 | | One Of My Tunes |
| B5 | | Don't Leave Me Now |
| B6 | | Empty Spaces |
| B7 | | Another Brick In The Wall (Part 3) |
| B8 | | Goodbye Cruel World |
| C1 | | Hey You |
| C2 | | Is There Anybody Out There? |

Fig. 2. Release Information on Demand

of black lines on the bar are used to indicate the position of current releases displayed with respect to all of them belonging to the Dot, hence guiding the user to the invisible releases.

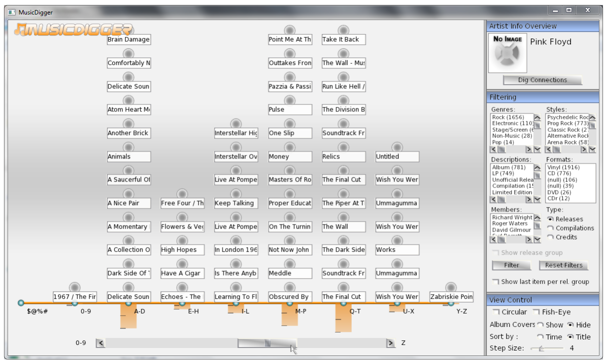
*4) Release Information:* A double click on the release displays the release information window where the description, genre, style, year, format, record label of the release are displayed. This feature is shown in Figure 2.

*5) View Controls:* The time-line can be switched between year and title by changing an option in the view controls. There is also an option to hide the album arts which can help in seeing more number of releases on the screen (Figure 3(a)). The graphical display can be changed to circular or fish-eye view based on user's choices. The circular effect renders a 3D look to the time-line as shown in Figure 3(b) and the fish-eye effect as shown in Figure 3(c), zooms in to the currently focused Dot while scrolling the time-line. Fish-eye view allows the users to get a closer as well as holistic view at the same time.

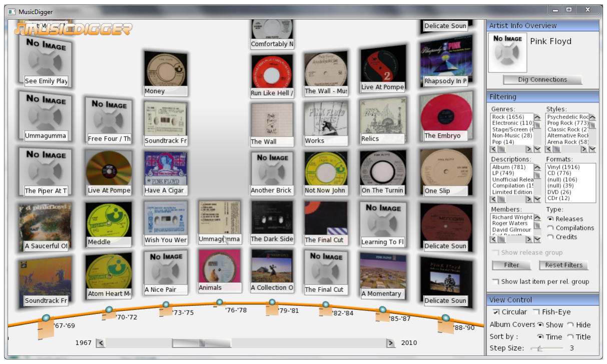
*6) Filtering:* The displayed releases can be filtered based on styles, genres, members, formats, descriptions, and item type. Users can select multiple options from various categories to narrow down on the releases they are looking for. The filtering panels are populated in the decreasing order of total number of releases associated with a type to give an indication of the kind of genres or styles that the current artist is mainly involved in (Figure 4(a)).

An album can be re-released many times over the years with various modifications to tracks, formats etc. and it becomes to difficult to search through the releases when there is a large number of such re-releases. So in the initial view only the first release associated with an album is displayed and an option to show the corresponding release group is provided. The user can select a particular release, enable the "Show Release Group" option and then filter to display the corresponding re-releases (Figure 4(b)).

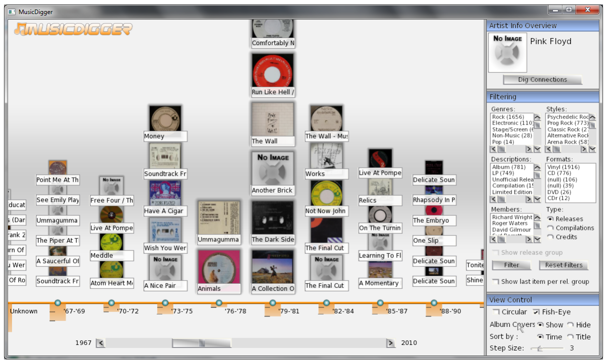
*7) Switching to Network Visualization:* The window on the top-right provides information about the Artist/Label whose releases are currently displayed. There is also a button which allows the user to easily switch to network visualization for the current Artist/Label.



(a) Sorted by Title and Album Arts are Hidden

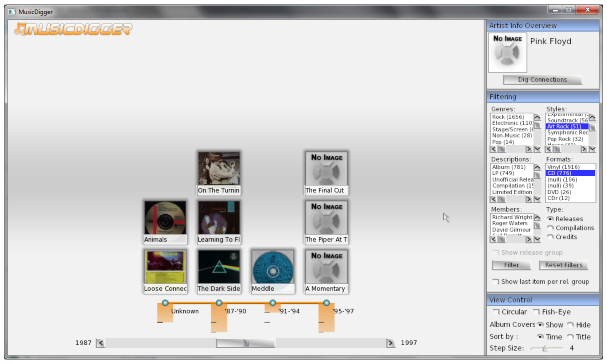


(b) Circular View

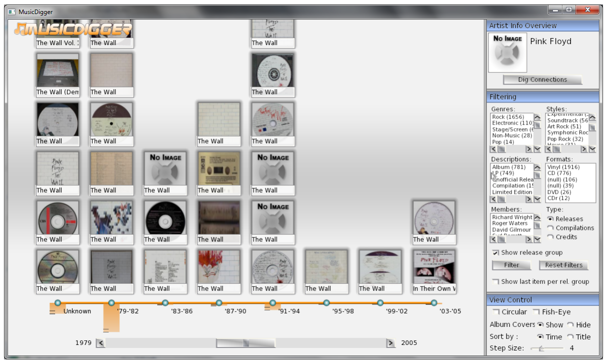


(c) Fish-eye View

Fig. 3. View Controls



(a) Filtering on Style and Format



(b) Displaying Release Group

Fig. 4. Filtering Options

## B. *Network Visualization*

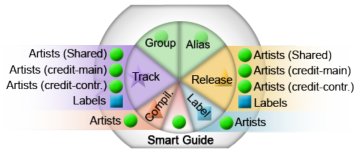
The network visualization as shown in Figure 6 allows the user to view the connections of an artist. After a successful search of an artist, the network visualization will show a node representing the artist. A context menu as show in Figure 5 that allows the user to explore the connections appears by clicking on the node. The network can be explored in various ways: other aliases which the artist goes by, releases that the artist has, groups the artist is a part of, labels the artist has produced releases with, tracks the artist has contributed to, and compilations the artist has contributed to. In addition to this there is a button in the artist info overview window that allows the user to expand the connections graph to artists who share the same band as the artist we are currently viewing. The view control window displays the legends for artists, labels, releases, compilations and tracks. as presented in the lower right panel of the window.

## V. Design and Implementation

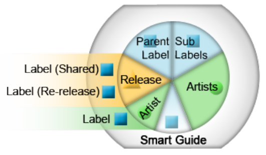
MusicDigger is written in C++ containing various modules to perform tasks like parse XML files, populate a database using Discogs data, query the database and parse the retrieved information, request release information and images in the background and finally the graphics rendering system.

## A. *XML Parser*

The data dump from DISCOGS is available in the XML format and had to be parsed for various schemas of the artist, label and release structures for extracting the information. We used a stream based parser, Expat, to parse the XML sequentially, since the large size of the data disallowed the use of a traditional tree based parser. Chunks of data were sent to the parser which would store the state of the parse elements and would then use this preserved state in parsing of the sequential buffered data. The data stream was encoded in UTF-8 format and the parser had to skip illegal characters. The parser is analogous to a plug-in service. This module is used both for populating the database and also during the runtime to retrieve the track information.



(a) Artist Node Context



(b) Label Node Context

Fig. 5. Context Menu Options for Network Expansion

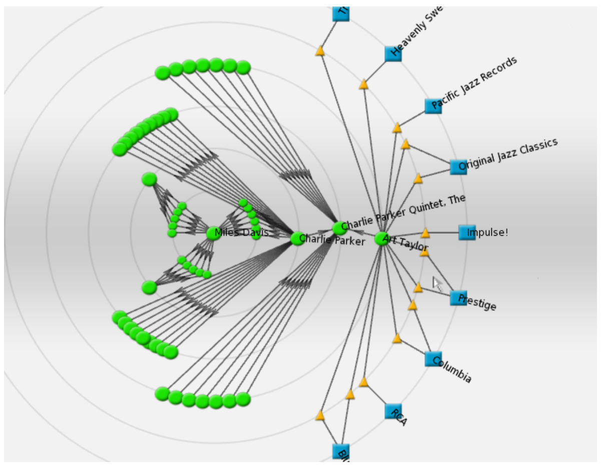


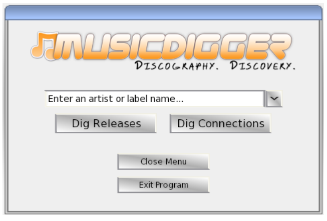
Fig. 6. A sample network visualization showing a 6-step exploration

## B. *Database and Interface*

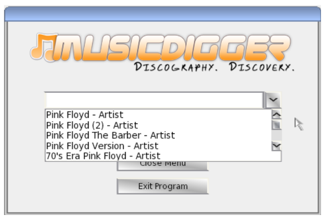
The database was designed to consist a sufficient number of tables with a minimum amount of redundancy and still support easy data retrieval for various features implemented in the application. The database schema used for the application can be found in the Appendix. The data parsed from the XMLs was populated in an SQL database using the SQlite API which provided versatile features to handle the large amount of data in a relatively small amount of time. We have tried to strike a balance between locally stored data on the computer and the data retrieved from the web while using the application. Information such as album art images, track names are not stored locally as they consume a huge amount of space, and are hence retrieved from web using the DataMiner module while running the application. However the application is still functional using only the local database without any connection to the web. The current database takes around 2GB of space which can be easily shared or transferred in a smaller zipped format which is around 600MB.

The information retrieval from the database module mainly contains three types of retrieval functions:

1) Search - The main search functionality is invoked when the user searches for a term in the search field. All artists/labels which match the term exactly (limited to the first 10 exact matches) and which are most similar to the search term (limited to the first 40 matches for labels and artists each) are retrieved and displayed as shown in Figure 7.
2) Attribute queries - These include queries about an artist's name, releases, aliases (different names of the same artist), bands, band members, releases, styles, genres, labels that the artist has released music under, and other information associated with an artist. For a release, these are queries about the release title, complete release information (masterID, styles, genres, year in which



(a) Default Start-up Window



Pink Floyd - Artist
Pink Floyd (2) - Artist
Pink Floyd The Barber - Artist
Pink Floyd Version - Artist
70's Era Pink Floyd - Artist

(b) Displaying search results

Fig. 7. Search Window

it was released, the labels it was released under). For a label, this information includes the releases released under that label, the styles and genres of those releases, the artists who released their music under that label etc.

3) Network link information queries - These queries form the basic framework for the network visualization. These serve to establish links like artist-artist, group - artist - group, artist - release - label, artist - release - artist, artist - track - artist, label - release - label, label - artists and so on.
4) Filtering query - We allow users to filter the releases based on various attributes like Genres, Styles, Formats, Item types, Members etc. The state of selected variables is captured every time the filter button is pressed and a nested query is constructed to retrieve the filtered releases from the database. A view of the filters can be found in the right hand panel of the screens displayed in Figure 1.

### *C. DataMiner*

This module makes use of the libcurl library to access the Discogs web site while the user is exploring releases. Dataminer is invoked as soon as the user double-clicks on a release to display the information in more detail as shown in Figure 2 or when a release becomes visible in the timeline. The release information obtained from the web is saved into the memory and passed to the XML parser to extract the track and credits information.

The Discogs API limits us to 5000 release requests daily so a lot of the code in DataMiner keeps track of number of release requests that have been made to ensure that an error is returned once we pass the maximum daily limit. In addition to this we have optimized the code to save the release requests made to a cache so that if a user decides to return to a release he/she looked at earlier we can pull it up more efficiently from the file system. This makes our application run more efficiently since a user is likely to search the same items again.

### *D. Rendering & Graphics*

MusicDigger visual interface runs on top of modern OpenGL[6], a cross-platform 3D rendering API, and uses OpeREng [7] library as the framework/engine for rendering management. The customized elements in left-side of the window, both in timeline and network visualizations, are managed using scene graph provided by OpenREng. Selection and visibility queries are based on scene graph components. Because of restricted camera and scene movements, some visibility options are implemented on application state, such as culling based on focused dot and release-list position of the time-dot. Text and other GUI elements such as buttons and listboxes are implemented using CEGUI library[7], which has its own 2D scene graph structure and its components are rendered into 3D windows using additional wrappers. MusicDigger manages these two different scene graphs synchronously, distributing events and adjusting rendering orders appropriately.

[6]http://www.opengl.org/

[7]http://cegui.org.uk/

## VI. Evaluation

The effectiveness of MusicDigger was evaluated by performing a usability test with six people. We came up with a form with three sections: A pre-evaluation questionnaire, a set of tasks to complete, and a post-evaluation questionnaire.

The pre-evaluation questionnaire was used to split users into three groups: music experts, music hobbyists, and casual listeners. We split users into three different groups as we wanted to determine impact of an application like MusicDigger among different kinds of people and would also indicate how widely adopted the application would be if it was ever commercially released. We also wanted to group these results as three different case studies. Among the questions we asked users in this questionnaire was their level of involvement in music and how active they were on web sites such as Discogs. We determined that music experts were individuals who were either music professionals or made heavy use of web sites such as Discogs, music hobbyists were individuals who made music a hobby but were not professionals and made minimal use of web sites listed in the pre-evaluation questionnaire, and casual listeners were individuals who only listened to music occasionally and were not familiar with the web sites listed in the pre-evaluation questionnaire. In the end we tested two music experts (a professional musician and radio programmer/music researcher), two music hobbyists, and two casual listeners.

The tasks of the user evaluation were laid out such that each user was required to use all of the functionality within a specific prototype of the application to complete the tasks. At the time the user evaluation was designed and conducted, all the features weren't implemented in the application and hence the tasks were designed to show us how easy it was for users to use the features currently implemented in the prototype. If any task was too difficult the user could skip it and he/she was encouraged to write about it in the post-evaluation questionnaire. Users were asked to complete the set of tasks after watching a short video that displayed all of the features we intended for the full MusicDigger application.

The post-evaluation questionnaire had three open-ended questions in which the user was asked to tell us the features of the application which they liked and worked well, the features that did not work so well, and also the features which they would like to see in the next release of the application. Then there were a series of questions in which users were asked to discuss their experience completing the tasks using the MusicDigger application by rating the features of the application, the speed and responsiveness of the application, aesthetic appeal of the application, and how easy it was to view and understand the data being presented.

For more detailed information on the user evaluation and how the actual form was presented look at Appendix.

## A. Case 1: Music Experts

The two music experts will be referred to as E.U. and D.A. We could not meet with E.U. and D.A. in person so they were unable to do the complete user evaluation, namely the section with the tasks. We conducted these sessions as more of an interview, asking them for their feedback and comments after showing them the video of features built into the application. E.U. is a professional musician and D.A. is a radio programmer and music researcher. Both E.U. and D.A. were familiar with the web sites listed in the pre-evaluation questionnaire (Pandora, Discogs, and Last.fm) and D.A. expressed to us that she uses these web sites very frequently. When we discussed the idea of MusicDigger with them they were both very enthusiastic about it. D.A. commented the following: "The ideas are incredible. I can access information visually easily, and it is right before my eyes." and E.U. expressed: "I would love to see this product as a more polished and commercial product. I would buy it myself for a few dollars and use it to browse discogs data." The two of them also provided very useful suggestions that are summarized below.

*Results:*

- E.U. found the network visualization to be more interesting than the timeline visualization. He liked the circular layout algorithm and expressed that music has many connections so it was nice to see this reflected in the network visualization. He thought it was intuive as well because when he browses Discogs he sees things in his mind as "sets of connected items" and this is reflected in the network visualization.
- He expressed that the access to release information was fast and that he liked our method of presenting multiple releases to the user and how the release opens up when you click it.
- He felt the visual aesthetic could be improved because it does not look very professional and he expressed a desire to filter by ratings/releases and country in the timeline. These features had not been implemented yet in the version of the application he saw.
- He also commented that he found the application very user-friendly, that adding the feature to view labels in the timeline visualization should not cause any problems and should be as efficient as it is for artists, and that he liked the idea of a smart guide to get the "most important" information but did not see how it worked because it had not been implemented.
- D.A. thought the year-based timeline visualization was a great way to discover an artist and thought that features such as filtering by style were very useful.
- She found that the album details appear very nicely and the information presented in the timeline was useful and clear to understand.
- She preferred time-sorting to alphabetic-sorting in the timeline but thought that alphabetic-sorting could be helpful at times so she was pleased to have the option to switch between the two.
- She appreciated the multiple filtering options in the timeline and expressed that it would make her "job easier in the future."
- She liked the concept of discovering artist connections in the network visualization and commented that it would make searching much easier.
- She thought that the network visualization had lots of options for those who wanted to access more but also felt that the visualization might be complex when a lot of information is presented on the screen.
- She seemed to have no preference using the circular or fish-eye views over the standard view option in the timeline.
- She felt that we could present more information and content to guide the user more. She commented that this information could include ratings, reviews, biographic information, and more textual content. She expressed that she used allmusic.com more than Discogs because of added features such as ratings and reviews.
- She liked the fact that this application did not require an internet connection to be used and mentioned that it would help her browse music when she is not connected to the internet.
- She expressed that for her work she sometimes prepares a radio session based on a single label or artist and that MusicDigger could help her decide and discover what items are similar, how they are connected to each other, and present details about songs and realeases in a highly accessible format. To introduce a song on the radio, one needs to conduct previous research and she found our tool very well-suited for the job.

## B. Case 2: Music Hobbyists

The two music hobbyists will be referred to as J.M. and J.A. J.M. and J.A. were both familiar with Last.fm and Pandora but not Discogs. J.M. spent ten hours a week using those web sites while J.A. spent less than five. Neither of them contributed to the web sites above and they both played the guitar. We met with the two participants in person so they were able to do the entire user evaluation. We found that neither participant was able to complete all of the tasks. J.M. struggled to complete taks 4 and 6 while J.A. struggled to complete task 7. We concluded that these tasks came later in the form and they were made progressively harder, which is why the participants struggled. We also took this to mean that there are some areas in which we could improve to make the more complicated features of our application (two-step searches in the network visualization and expanding artists with shared bands) easier to understand. They left us with some comments summarized below.

*Results:*

- J.A. felt that the information presented was available very quickly and that it was thoroughly laid out. He thought all of the features of the application were good but some of them were difficult to use, such as discovering where the expanding artists with shared bands button was.

- He suggested we add in album artwork and artist pictures because that feature had not been implemented yet.
- J.A. also commented that the information available was accurate but at times it was presented in a disorganized manner. He also expressed that he had some trouble switching between displays, scrolling, and that he did not like the color scheme we chose.
- J.A. found all of the features in the application to be helpful and also liked the graphical aspects.
- J.M. liked the organization in the timeline visualization and found it to be very helpful to discover new music from bands he was less familiar with. He liked sorting by time and by title.
- J.M. expressed that he did not like the network visualization but elaborated by saying he thought this visualization was less useful for him and that maybe a music professional would find it helpful. He also found the visualization to be a little less intuitive.
- J.M. mentioned that he was only interested in co-artist and band/group information and did not need the other features of the application.
- He suggested that some additional features for the application could be to play tracks directly (if possible), integrate youtube/wikipedia/google search on tracks/artists/band names, include the lyrics of songs, include the reviews of songs, and to be able to use the filters to get results based on these reviews.
- J.M. found the information presented to be accurate but overwhelming at times, especially when using the network visualization. He also expressed the need for better search and filtering since he used an earlier iteration of the application code. He thought the application performed slowly and that the scrolling and display switching mechanisms were average.
- J.M. found the timeline visualization, view control windows, overview window, and control panel to be very helpful when completing the tasks.

### C. Case 3: Casual Listeners

The two casual listeners will be referred to as H.V. and F.T. H.V. was only familiar with Pandora while F.T. was familiar with Last.fm and Pandora. Both participants spent less than five hours a week on these web sites and never contributed to them. We were able to meet with H.V. in person so she tried her hand at completing the tasks but F.T. did not do the tasks. H.V. was able to complete all of the tasks but did start to struggle with the later tasks that required the user to use the more complicated features of the prototype she evaluted. The two participants gave us some useful feedback summarized below.

*Results:*

- H.V. thought all the features of the application were nice but she especially liked the circular menu in the network visualization.
- She thought the expand artists with shared bands button could be placed in the circular menu so that it would be more intuitive and easier to access. She also felt that the circular menu had too many options so it could be simplified and that some of the options were not intuitive.
- She thought that the information presented was accurate and that the application had a fast response time.
- She felt that scrolling vertically in the timeline visualization was not as easy as it could be and that it was hard at first to know that the logo button could be used to go back to the welcome screen.
- She liked all of the visual display elements of the application and found all of the features she used when completing the tasks to be very helpful.
- F.T. liked the idea of the timeline visualization, its filtering options, and the bar chart.
- F.T. thought that the network visualization was more impressive than the timeline visualization, the coloring made it easy to read, the information presented was uncluttered and easy to understand, and that the layout algorithm seemed intuitive.
- He suggested that we use different colors to specify whether an artist node was a group or an individual.
- He also suggested that the MusicDigger idea could be applied to movie databases as well and that it might be interesting to add a time-dimension to the network visualization.
- F.T. thought the information presented was accurate and the the application was fast. He also liked all of the visual display elements of the application and felt that all the features he had a chance to use were very helpful.

## VII. Conclusions and Future work

Based on our literature survey and overview of existing tools, we think that MusicDigger offers unique opportunities to discover music in a visually direct, attractive and easy way, based only on metadata. Our two basic interface solutions offer features that complement each other, while switching between the views can be done in a single click. Responses from users with different backgrounds shared the fact that the interfaces are found easy to use, understand and interact with, and the features are found appropriate for most exploration scenarios. Many of the users has shown their interest for a more polished and finished product, or support for other datasets, hoping to use it for their own exploration in near future.

Although we were able to implement most of our ideas for browsing and discovery, we couldn't complete, or evaluate in depth, some features that we didn't have the time to implement or experiment in the limited time frame.

Firstly, our system can be improved with features for guiding purposes. Currently, node size adjustment according to release count attribute is a step towards this goal. Likewise, some connection options that are likely to serve as recommendations could not be completed, including compilation links between artists, and the smart-guide feature, which was conceptually planned to run all connection queries at once and return all the links between artists in a priority sorted order. We should note that these features carry challenging implementation problems because of data size and query speed restrictions.

Although we have a large set of potential expert users within discogs community or record collectors, we could not increase our presence and run an evaluation with external users from this community. 3 external people showed their interest through our facebook page, yet did not volunteer for a user study. Running an evaluation with a larger group of knowledgeable users will shine more light on our approach and bring up new interesting ideas for our tool. A complete user study can also be based on overall satisfaction over a long term use, based on the number of items they discover, their trust in the results we display and their preference to use our tool instead of the web interface offered by discogs.com.

As for improvements of current features, the highest impact can be achieved by implementing an advanced layout algorithm for network visualization, which uses space more efficiently by adjusting arc size according to number of releases, along with other improvements. We observed that for queries that return a high number of results, network becomes increasingly hard or impossible to observe. Our filtering options are based on mitigating this problem, and there is a large room for improvement using simple yet effective filtering options. Scalability can also be improved by defining nodes that hold multiple items, along with easy to use, yet powerful management of such multi-item nodes.

The size and query speed of the large offline database has become a problem as more complex functionalities are implemented and as our data source keeps growing. A scalable approach can be to have a small online server cluster as a part of the system to send responses, with a focus on short response time, but the power of offline access capability would be lost in this case. Another approach is to create subsets of the big database based on user preferences. For example, a user may only be interested in jazz genre, and so can use a subset of data which does not include artists not related to jazz, and their releases. There is also room for many small improvements for increasing the ease of use and improving the presented information. One such feature is extending search results with a few highlights of artist information along with artist name, such as most related music style, the year of the first release, or total number of releases.

As final words, we should note that digging music is not complete without the ability to hear the sounds, feel the vibes and live the ideas. Our best hope with MusicDigger is to help others find their way easier in the vast jungle of music into the gems that they would love to know more about.

## VIII. Acknowledgements

We would like to extend a sincere thanks to Robert Gibson, the Director of the University of Maryland School of Music, for being our faculty sponsor on this project and taking the time to meet with us this semester. His enthusiasm, support, and feedback helped us greatly improve the MusicDigger application. We hope that he and other faculty members in School of Music will make use of it in future courses, which also studies the works of artists as discografies.

We would also like to thank Dr. Ben Shneiderman for providing us with valuable information as we studied the material presented in his information visualization course and for giving us great advice and feedback as we worked to complete this project. For the same reasons, we would like to thank Gleneesha Johnson, the teaching assistant for this course.

Lastly, but certainly not least, we would like to thank all who graciously donated their time to complete our user evaluation form and gave us the feedback we needed to make this project a success. The critiqiues we received from our friends in class also helped us gain a better understanding of our approach and also served as example user evaluations for us. With their acknowledgement, we are presenting an incomplete list of people who evaluated our software as the following: Evren Ulusoy, Ahmet Gunes, Duygu Ates, Ferhan Ture, Helvia Velloso, and Jonah Askarinam.

## References

[1] Piotr D. Adamczyk. Seeing sounds: exploring musical social networks. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 512–515, New York, NY, USA, 2004. ACM.

[2] Raimund Dachselt and Mathias Frisch. Mambo: a facet-based zoomable music browser. In *Proceedings of the 6th international conference on Mobile and ubiquitous multimedia*, MUM '07, pages 110–117, New York, NY, USA, 2007. ACM.

[3] Ricardo Dias and Manuel J. Fonseca. Muvis: an application for interactive exploration of large music collections. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1043–1046, New York, NY, USA, 2010. ACM.

[4] Anita Shen Lillie. Musicbox: Navigating the space of your music. Master's thesis, Massachusets Institute of Technology, 2008.

[5] Mark Notess. Variations2: Toward visual interfaces for digital music libraries. In *Second International Workshop on Visual Interfaces to Digital Libraries*, 2002.

[6] Marc Torrens and Josep llus Arcos. Visualizing and exploring personal music libraries. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, pages 421–424, 2004.

[7] M. Adil Yalçın. Open rendering engine. http://openreng.sourceforge.net/.

## Appendix

### A. Database Schema

Artist Table

1) int Artist ID (1 reserved for VARIOUS)
2) string Artist Name
3) string Image Url
4) int parentAliasID

Artist Member table

1) int groupID
2) int artistID

Label Info table

1) int Label ID
2) string Label Name
3) string ImageURL
4) int Parent ID

Release Main Table

1) int Release ID (assigned by DISCOGS)
2) string Release name
3) string Album Art url
4) int Master ID (not an actual release id)
5) int year

Release Genre Table

1) int Release ID
2) enum Genre ID

Release Styles Table

1) int Release ID
2) enum Style ID

MainArtist Release Table

1) int Release ID
2) int Artist ID

ExtraArtist ReleaseTrack Table

1) int Release ID
2) int Artist ID
3) string trackPosition
4) string/enum role

Compilations Table

1) int ReleaseID
2) int ArtistID
3) string trackPosition

Label Release table

1) int Release ID
2) int Label ID
3) string Catalog Number

Release Format table

1) int releaseID
2) int format
3) int quantity
4) int description

### B. User Evaluation Form

*Pre-Evaluation Questionnaire:* Answer the following questions to help us understand your involvement in music. If any question makes you feel uncomfortable feel free to skip it.

1) What are some artists you enjoy listening to? Separate multiple artists by commas.
2) What are some genres of music that you like? Separate multiple genres by commas.
3) Which of the following web sites do you know about? (Circle each one that applies)
   a) http://www.discogs.com/
   b) http://www.last.fm/
   c) http://www.pandora.com/
4) If you circled web sites above, how many hours per week do you spend using them? (Circle only one)
   a) Less than 5
   b) 5
   c) 10
   d) 15
   e) 20
   f) 25
   g) More than 25
5) If you circled web sites above, how often do you contribute to them? (Circle only one)
   a) Never
   b) Rarely
   c) Yearly
   d) Monthly
   e) Weekly
   f) Daily
   g) Other: _________________________
6) What is your level of involvement in music? (Circle only one)
   a) Casual listener
   b) Music hobbyist
   c) Professional musician
   d) Active contributor to a music web site, e.g. www.discogs.com
   e) Other: _________________________
7) Do you play an instrument? If so, list it below otherwise leave it blank. (If you play multiple instruments you can separate them with commas)

*Tasks:* The facilitator will now show you a short video demo of the application and will answer any questions you might have. After that period, you will have 30 minutes to complete the following tasks. If the evaluation is being conducted online skip this section.

1) Use the timeline visualization to find a release of the album Killem All by the band Metallica that occurred after 1983. Give the year of the release and the label separated by commas.
2) Use the timeline visualization to give the total number of releases the band Coldplay had in 1999.

3) Use the timeline visualization to list the titles of three releases by the band Nirvana. One must start with an E, the second must start with a K, and the last one must start with a U.
4) Use the network visualization to list the record labels the artist Dave Grohl has produced releases with.
5) Use the network visualization to find the total number of groups that Dave Grohl has been a part of.
6) Use the network visualization to first find the groups Edu Falaschi is a part of, then the total number of labels that group has produced releases with.
7) Use the network visualization to list the artists (by name) that share the same band as Dexter Holland.

*Post-Evaluation Questionnaire:*

1) What were the features that impressed you the most, both in the network and timeline visualizations?
2) What were the features you disliked and would prefer not having in the application, both in the network and timeline visualizations?
3) What additional features would you like to see in the next version of the application?
4) What was your experience with the information available? (Circle all that apply)
   - a) Accurate/Inaccurate
   - b) Just Enough/Overwhelming
   - c) Sufficient/Insufficient
   - d) Organized/Disorganized
   - e) Other: ________________________
5) How would you rate the speed/responsiveness of the application in accomplishing the tasks above? (Circle only one)
   - a) Very Fast
   - b) Fast
   - c) Moderate
   - d) Slow
   - e) Very slow
6) How was your experience with the following visual display elements? (Circle only one for each)
   - a) Screen resolution
     - i) Great
     - ii) Good
     - iii) Average
     - iv) Poor
     - v) Bad
   - b) Display colors
     - i) Great
     - ii) Good
     - iii) Average
     - iv) Poor
     - v) Bad
   - c) Scrolling and Other Feedback
     - i) Great
     - ii) Good
     - iii) Average
     - iv) Poor
     - v) Bad
   - d) Switching displays
     - i) Great
     - ii) Good
     - iii) Average
     - iv) Poor
     - v) Bad
7) How helpful did you find the following features? (Circle only one for each)
   - a) Timeline visualization
     - i) Very Helpful
     - ii) Helpful
     - iii) Unhelpful
   - b) Network visualization
     - i) Very Helpful
     - ii) Helpful
     - iii) Unhelpful
   - c) View control windows
     - i) Very Helpful
     - ii) Helpful
     - iii) Unhelpful
   - d) Overview window
     - i) Very Helpful
     - ii) Helpful
     - iii) Unhelpful
   - e) Filtering window
     - i) Very Helpful
     - ii) Helpful
     - iii) Unhelpful
   - f) Control panel
     - i) Very Helpful
     - ii) Helpful
     - iii) Unhelpful