

Trax: Better Mp3 Organizing Through Visualization

Dave Ternes

Department of Computer Science
University of British Columbia
201-2366
Main Mall
Vancouver BC Canada V6T 1Z4
drternes@cs.ubc.ca

ABSTRACT

Mp3s are increasingly the dominant format for recorded music. However, the current organizational methods for mp3 collections are lacking. Thus I propose the Trax system, a visualization tool for organizing mp3s according to genre, and for creating playlists. Trax works on the assumption that most songs exist in multiple genres, and thus creates a graph representing these links between different genres. Trax provides many different ways of dealing with the complexity of the genre graph, allowing for viewing of details and overviews, and all the while supporting the creation of playlists. The results of the system are encouraging, in that it has met all initial design goals. However, Trax has also provoked several larger issue to do with musical genre, which is both a positive and negative aspect of the system.

1 INTRODUCTION

“Organizing CDs or records is a *lot* more fun than organizing computer files; that's just a proven fact!”

-Chuck Eddy, Senior Music Editor, *The Village Voice*

To obsess over music is not just to obsess over the sound of music itself, but also inevitably it is to fetishize the physical mediums of recorded music. Beyond any claims of analog superiority, one of the main reasons for the record's longevity in terms of music-lovers' interest is its sheer physical presence. The artwork is bigger and richer, and to look at the record itself is to actually *see* the music carved into its grooves. Even CDs still maintain this physical presence, presenting an object onto which one can project ones obsessions and enjoyment. Mp3s lack all of this presence, represented only by a string of text within some mp3 player, it is not surprising that their abstract nature have brought about complaints such as the quote displayed above. However, mp3s lack of physical presence has also lead to their widespread availability and current dominance of the music market. Even still, there continues to be a strong desire for some representation that catches the imagination and can develop and worthwhile interaction—witness the recent massive hype around the iPod.

The iPod succeeds by once more providing an attractive physical object to be the subject of our desires—in addition to being a very well designed piece of hardware, of course. But is it possible to stay within the digital realm and still create an interesting and worthwhile system for interacting with mp3 files? This paper presents an attempt to deal with this issue through the creation of the Trax system for sorting, viewing and playing a collection of mp3 files. Trax's main tool in this is a node-link graph of all the genres contained within an mp3 collection, which can be traversed, filtered and added to, creating a unique

visualization of set of mp3 files. This system seeks to provide a satisfying and intuitive way of managing a large, personal mp3 collection by making use of all the advantages that a digital file on a modern PC can have over a stodgy physical object in real space, rather than try to emulate the appeal other physical objects.

The main user actions that this system aims to support are organizing and labelling mp3s and browsing an mp3 collection with the aim of creating playlists. The dataset used will be the authors own mp3 collection, comprised of rough 7000 files (50+GB) and about 3000 different artists. It is aimed at users who are fairly invested in their mp3 collection (and music in general) and would be willing to spend the time to properly organize their mp3s according to genres. Though the Trax system would still work fine for a small mp3 collection, its real benefit is in allowing people with large mp3 collections—who usually must spend a large amount of time browsing through files to find what they want—to quickly and easily browse through their files according to a sensible organizing principle.

2 RELATED WORK

Trax uses several well established visualization techniques to represent its information. Graph visualization is used for a wide variety of tasks. In particular the work of Jeffery Heer is highly relevant to this project [7]. He showed in his work using force-directed layout to display small world graphs for such things as the Friendster network the power of such simple graph layout representations [6]. Though many other layout algorithms are also available for displaying graphs [8] their applicability to the music genre datasets does not match that of forced-directed layout, as they do not take full advantage of the heavily interconnected nature of the genres.

In addition to laying out graphs, there is a wide range of tools for aiding and expanding how people view graphs. Zooming, panning, Fisheye lenses, Perspective Wall and Overviews are all means of providing differing views of a graph [5][8][10]. All of them can provide users with more detailed or useful information about a graph they are using, and have proven to be very useful in different situations [1][11][15].

Another related concept is focus+context, a very popular idea in the field of visualization. The goal of focus+context is allow users to zoom in on a particular detailed section of a graph (or any other visualization) while at the same time ensuring that the user maintains an overview of the entire system that he or she is viewing. Focus+context techniques have been successful applied to many diverse areas. Overall, these techniques have proven quite useful in dealing with complex visualizations—allowing them to retain their level of detail and complexity without being dumbed-down, and all the while maintaining usability [12].

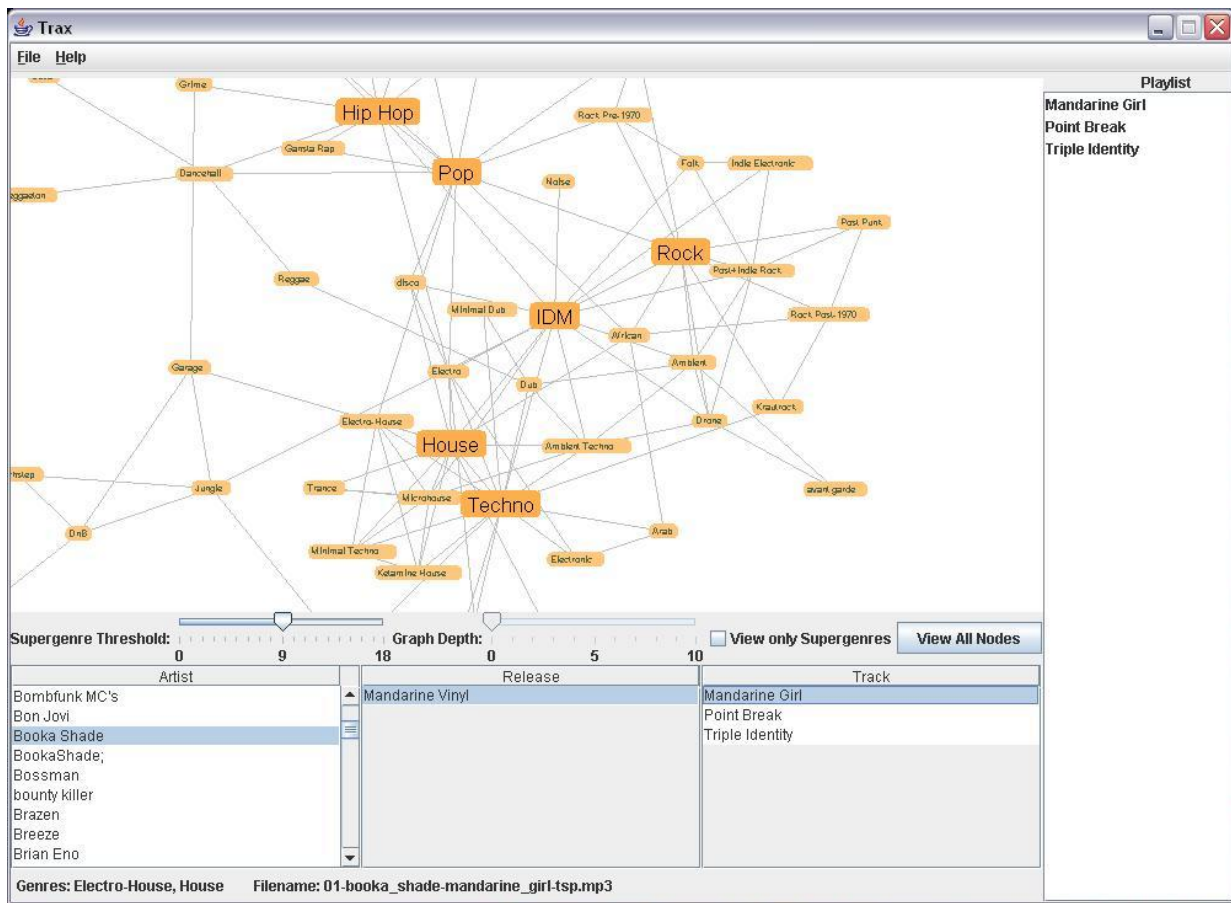


Figure 1. The Trax Interface

Somewhat surprisingly, there has seemingly been a relatively small amount of research into the application of visualization technology to music and peoples listening habits. Of what has been done, a large amount focuses on recommender systems and how to visualization the links from a given artist to other artist that a person might enjoy [14]. A more interesting line of research has been conducted by Eric Brochu, a PhD student at UBC. Brochu has developed a machine learning algorithm than can successfully determine mood and high-level genre of a song [3]. He and Heidi Lam have both been working on interesting ways of displaying the information produced by this algorithm, and have used in one instance a form of node link graph [9].

Finally, this paper makes reference to many genres of electronic music that may not be familiar to most readers of this paper. For an easy introduction to the myriad subgenres of electronic music, readers could do worse than to start at Ishkur's Guide to Electronic Music (<http://www.di.fm/edmguide/edmguide.html>), which has sound clips and brief descriptions of many different electronic genres—though the site is highly subjective and by no means complete in its overview. For a more serious investigation of electronic music and all its variations, the seminal text on the area is Simon Reynold's *Generation Ecstasy* which has an excellent overview of the development of the genre [13]. Also recommended are Brewster and Boughton's *Last Night a DJ Saved my Life* which recounts the history of the DJ and the development of dance music [2], as well as Kodwo Eshun's *More Brilliant than the Sun* which presents a more philosophical

overview of various music genres, including many forms of electronic music [4].

3 DESCRIPTION OF SOLUTION

The crucial element of the Trax system is the genre graph, a representation of all the genres that the mp3s of a given collection are in. What makes the genre graph a graph, and not just a set of isolated nodes, is the very straightforward concept that most songs do not exist in only one genre. Unless you use the highest-level possible genre terms like "Rock" or "Dance" it is very hard to classify a song as fitting neatly in only one genre, and then you lose much of the classifying power of the genre tag. Unfortunately, this is exactly all that you are allowed to do in the current version of the mp3 ID3 tag, and all mp3 players support only this level of labelling. Where Trax differs from other popular mp3 players like iTunes is that it allows the user to label a track with as many different genres as he or she wants. Thus users can label a Beach Boys song starting with the high-level genre of "Rock," and then narrow it down to "60's Rock" and then narrow it down again to "Surf Rock." Given such an mp3 file, Trax will create a node for each of the given genres, and creates links between all of them.

As can be seen then, the idea behind the graph is that for each mp3, for every genre it shares, a link is drawn between those genres. The strengths of the links are proportional to the number of mp3s that have both those genres. Thus the graph can be created using a force-directed layout, in which highly related

genres will be closely positioned and more unrelated genres will be farther apart. This is the overall driving idea behind the Trax system: a graph-based visualization of a set of mp3s based on the potentially multiple genres in which each mp3 resides. The rest of the system is based around making this visualization easy to interpret and navigate through, as well as ensuring that it is also useful for our actual users, and not just a pretty picture—which is the true meat of the system, and determines the success or failure of the overall idea.

3.1 Supergenres and Subgenres

One of the problems inherent to this approach is that of the subgenres-supergenre relationship. As has already been indicated above, genres often come in hierarchical relations to each other with genres such as “Rock” or “Techno” being very high-level, while genres such as “Post-Punk” or “Detroit Techno” are much more specific. What is troublesome for graph making is that there are far more subgenres than there are supergenres, which means that there will a handful of nodes with many links coming into them, with a large number of other nodes that will be not as heavily linked to. As a result of this graphs could become highly visually cluttered.

To help ameliorate this issue, Trax will support the concept of supergenres, and will indicate in the graphs through the colour and size of supergenres nodes. Supergenres will be larger and more saturated than other genres, and all links will also be less saturated in order to not demand too much visual attention while still remaining salient. This has the effect of creating visual popout for the supergenre nodes, and also allows them to be read clearly even when zoomed out, unlike the small, regular nodes. Furthermore, users will also be able to toggle between viewing all the genres, and viewing just the supergenres. When viewing just supergenres, all other nodes will not be visible, however the links to the genres will still be visible. This provides the user with a version of focus+context, where the user can more easily view the most popular nodes while still maintaining a sense of where they fit within the larger graph framework. This is an easy way of getting a high-level overview of a collection of mp3s.

Whether a node is a supergenre or not will be determined by whether it has a number of direct neighbours greater than a certain threshold. This threshold will be user controllable using a slider, though a reasonable default will be provided that classifies on average about a fifth of the nodes as supergenres.

3.2 Viewing a Single Genre

Even with implementing a supergenre-subgenre relationship with all the nodes, and allowing for zooming and panning, there are still likely to be many links crisscrossing the screen and many nodes throughout the graph that, at any given time will serve no purpose to the user other than creating visual clutter. That is why the Trax system allows users to view a graph centred on one genre which shows only its direct neighbours. Users can simply double click on any node to view a graph centred around that genre.

Users will be able to control the depth of the graph shown using a slider, so that the graph will display only the neighbours one link away, or two, or three, or whatever the user stipulates. In this way they will be able to focus in on one particular genre, and how it relates to others genres, without being distracted by unnecessary nodes and links. Furthermore, if the user wishes to see which of the neighbouring nodes itself branches off into different areas, they can either expand the depth of the graph by one, or select that genre to view in its own genre centred graph. At any point during the interaction, users can double click on another node to bring up

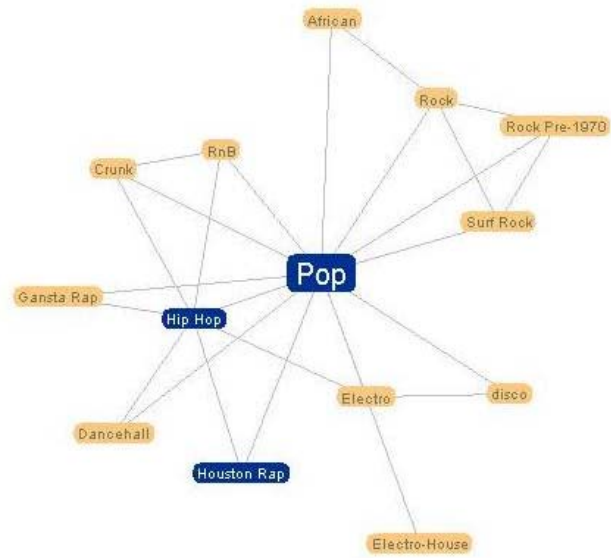


Table 1. A genre-centered graph of depth 1 with 3 nodes selected

a genre-centred graph of that genre, or return to the full, all-genre graph.

The supergenre-subgenre functionality is also present in the single genre-centred graph, but scaled to fit the small graph, so that the central genre always sits at the threshold for being a supergenre. Users will again have the ability to view just the supergenres, or all genres. In this way, the genre-centered graph functions simply as a smaller version of the larger, full-genre graph, with all the functionality it has, plus the ability to expand and contract the depth of the graph displayed.

3.3 Viewing, Editing and Adding Mp3s to Playlists

Trax is meant to be a full featured system for managing an mp3 collection, so obviously the above features are dependant on there being a way to actually see the mp3s that fall under each genre. This is accomplished mainly through having a dynamic, browseable list of the artists in whatever genres are selected. Each artist has their associated mp3s classified by their release and then track name, and are browseable as such. Genres can be selected at any point in time when viewing a graph, and also when viewing a genre-centred graph, the artists for that genre are listed by default. Further genres can be selected, narrowing down the list of artists to the intersection of whatever genres have been selected. When browsing through these artists, a list at the bottom of the screen will always display the genres in which they actually reside, as well as the actual file name for the mp3. Additionally, users can edit any of the artist, release, or song title information for any track.

Another important action that Trax supports is the labelling of mp3s with different genres. This can be done by labelling either an entire artist, release, or just a single mp3 file with one or more genres. Users can create new genres or use existing genres for their labelling. Adding tracks to existing genres can be done via a popup menu, that allows users to select any existing genre to add. Adding tracks to new genres can also be accomplished via popup menu. Labelling of genres can be done to individual tracks, or also to entire releases or artists, which will have the effect of going through and classifying all mp3s from the given release or

artist as being in that genre. The ability to do label genres—and to do so relatively efficiently—is key for allowing the user to build up a useful and interesting graph of their mp3 collection.

Lastly, Trax supports the creation of playlists using the mp3s in the system. At any point in time while browsing through an mp3 collection, a track, release, or an entire artist can be added to the playlist. Playlists can then be exported as .m3u files, a standard playlist file type readable by all major mp3 players including iTunes, Winamp, and Windows Media Player.

4 HIGH-LEVEL IMPLEMENTATION

The Trax system was implemented in Java, using the Prefuse library for graph drawing, and Swing for other GUI elements. It uses Prefuse's in-built force-feedback layout algorithm to create the genre graph. It also stores all mp3 information in a separate xml file to allow for easier access and manipulability

4.1 Reading in mp3s

The first task that Trax must do in order to be at all useful is read in a collection of mp3 files and parse them into a usable format. The reading of mp3's ID3 tags is accomplished using the JID3 library (available at <http://jid3.blinkenlights.org/>) which reads ID3v1 and v2 tags. The Artist, Title, Album, Genre and full path name were then written to an xml file that could later be used to build the graph. The writing and reading of the xml file was accomplished using the Xerces java library [LINK].

Since many mp3s are not fully and accurately tagged and, most importantly, the genre tag is rarely properly used, if used at all, several compromises needed to be made in how the system reads in mp3s. For mp3s without ID3 tags at all, they were not added to the system, as parsing their filenames to discover some sort of meaningful information was beyond the scope of the project. For files with incomplete ID3 tags, whatever information was missing was simply replaced with an "Unknown" value. This resulted in a large number of "Unknown Artist" mp3s, but no easier solution could be found. Additional irregularities in the naming of certain artists, such as added spaces and punctuation, often caused there to be duplicate artist entries for the same artist. A more complex and robust parsing system for the mp3 is definitely desired, but was not implemented due to time constraints.

Finally, the issue of the genre tags was quite troublesome. In my own experience, using my own mp3 collection, I found that the genre tags were almost completely useless, with tracks rarely even having any genre listed, let alone an accurate one. Fortunately, I had my collection already organized in folders by broad, overall genre. Thus I decided to use the folder name in which an mp3 resides in place of its genre, for the initial creation of the mp3 list. This worked particularly well in the case of the original dataset, but would also work reasonably well in an mp3 collection sorted into folders by artist, in that it would be fairly easy to then assign each artist node a genre, and then proceed from there. Though this is not ideal, again, no better means of classification was available.

4.2 Creating the Genre Graph and Initial Display

To create a new genre graph, first the xml file for the mp3 collection is read in and all of the track information is stored in a sorted list by an artist class. Each artist is an individual class which has a link to a list of their releases, and each release has a link to its tracks. The tracks themselves contain all the information the associated mp3 has, including the Title, Genres, and full mp3 path. When information is added to the track, such as a new genre, it automatically updates all the relevant fields in the

related release and artist classes, so that the overall data structure never gets out of sync.

Once the list of mp3s has been fully created, the list is parsed. For each track, a node is created for every genre it has (if it has not already been done so), and a link is drawn connecting each of these nodes to each other. Each node is also assigned a weight function proportional to the number of mp3s that are in that particular genre. Additionally, as the graph is built up, the number of neighbours that a particular node has is continually checked. If a node crosses the threshold, which is initially set at half the number of the maximum number of neighbours any node has, then that node is assigned as a supergenre, using the attributes function of Prefuse's NodeItems. This attribute can then be later checked using custom colour and size function to change these nodes' appearance relative to normal genre nodes.

Finally this graph is built up using Prefuse's inbuilt graph drawing functions, using a straightforward filter that simply displays all nodes and edges. Trax also uses Prefuse's basic force-directed graph layout algorithm, using anti-gravity forces on the nodes, as well as a spring force on the links.

Once the graph has been built the artist list is populated. Initially, all artists for all genres are added to the list; however, at any time users may select a single genre or a group of genres to view their associated artists. This is accomplished using three interlinked functions, one each for artist, release and track, that parse the artist list and its associated release and track lists, filtering for the required genres.

4.3 Drawing a Genre-Centred Graph

Trax also has a second graph view, in which it displays a graph centered around a single node. This required a unique graph filter for displaying the graph in this format. The nodes to display for a genre-centred are calculated using a recursive function that behaves in a similar fashion to a tree traversal. The function is provided with a depth to process to and a genre to centre on, after which it recursively traverses each node that is a neighbour of the specified genre and adds it to the graph. Unfortunately, the genre graph is not a tree, and is in fact heavily interlinked. This results in a constant checking of each new node traversed to ensure that it has not already been added to the graph. Though it is unclear what search algorithm is used by Prefuse's basic membership-checking function, it seems likely that is it at least somewhat costly, as there begins to be a noticeable delay after moving to higher depths. Fortunately, at higher-depths it is just as easy to view the full graph, which is much more quickly rendered.

5 RESULTS

Due to time constraints, a formal evaluation of the Trax system could not be completed in any detail. Some high-level feedback was sought from a handful of colleagues, but their suggestions, while interesting, were not enough to provide a solid determination of the success of the system. Consequently most of the analysis of Trax is thus based on reasoned argument from my own experience using the system. This is not necessarily a complete failure, however, as the Trax system was designed to a large part to meet my own needs as serious music enthusiast. The dataset used was my own mp3 collection, so thus there was definite real-world buy in to my own use of the system. Of course the fact that I am also the designer of the system greatly biases the results, but due to the fact that I actually legitimately can be classified as a target user, my assessment of the system is at least not based my hypothetical projection of a user's needs, of which I may not truly know anything about.

5.1 Scenarios of use

Though the following scenarios are written in the classical scenario third-person style, they also accurately reflect my own experiences using Trax, and represent the two most frequent and important tasks that I performed with the system.

5.1.1 Exploring an Mp3 collection

One of the main uses of the Trax system is to allow a user to explore their collection of mp3s and see new and interesting connections amongst tracks and artists. Trax supports these actions through various interaction techniques centred around the genre graph. Here is a hypothetical scenario of use along these lines:

Simon is feeling like listening to some Grime, so he fires up the Trax system and is presented with the main page, displaying a large graph of genres. Simon, being the expert on music that he is, knows that Grime is related to Hip Hop. In this case, Hip Hop is highlighted as a supergenre, so it is easy to spot, but just to make it even easier, Simon, clicks the “View only supergenre” checkbox to hide all other nodes. At this point, he spots the node easily and double-clicks on it in order to see a genre-centred graph of Hip Hop.

A new graph is then displayed with Hip Hop in the centre, and all the genres that are direct neighbours of Hip Hop. In the artist list view, all of the Hip Hop artists are listed. Simon is quickly spots the Grime node and double clicks on it, bringing him to a genre-centred graph for Grime. Simon peruses through the artists for Grime, and enqueues all of his tracks by Wiley, as well as Ruff Squad’s “All Day Long.” Then he notices that there is a link from Grime to Houston Rap, which reminds him that there has been some collaborations between these two, geographically quite separated scenes. He control clicks on the Houston Rap node, and in the artist list view is displayed all artists who exist within both genres. He sees that there is a collaboration between Grime MC Dizzee Rascal and Houston crew the Grit Boys so he enqueues that as well. Thinking of Houston Rap has made him hungry for more, so he double-clicks on Houston Rap, and enqueues several tracks by Mike Jones, as well as Paul Wall’s “Sittin’ Sidewayz.”

Satisfied with his new playlist, Simon chooses to export the file as an .m3u file by selected the “Export playlist” option from the file menu and selecting the destination for the file. Simon can now go and listen to his playlist on the mp3 player of his choice.

5.1.2 Organizing an Mp3 collection

In order to get to the stage where a user can explore his or her music collection, the user must first organize the collection and label the mp3s according to genre. Here is a scenario of use for such a purpose:

Jessica has just gotten the Trax system and is eager to start organizing her files with it. She is an avid music fan with a large collection of mp3s who spends a lot of time sorting through them trying to find the right track. Consequently she is willing to commit a fair amount of time to getting things working with Trax. Jessica opens up Trax and selects “Import mp3s” from the File menu, and selects the directory where she stores all her music. Trax reads all of the mp3s and uses the parent directory of each mp3 as their genres, in order to give all mp3s an initial genre. Jessica has her mp3s sorted largely by artist, with a few folders for miscellaneous tracks for different genres. Once the import is complete, she is presented with a series of unconnected nodes and a list of all the artists in her collection. Diligently, she goes through and assigns all artists a high level genre like Rock, Pop, or House, as well as a few more specific subgenres, by selecting each artist and choosing “Add to existing genre” or “Add to new

genre” from the popup. After this process (which does take a while) she has a very basic graph with a few links between them. Now she is able to view single genres, filter out many of the extraneous artist and getting the artist list down to a manageable size. From this point, Jessica starts to enjoy herself, as she peruses through her mp3 collection pondering the genre placement of many of her favourite artists. New genres are created and new connections made, and the graph slowly grows. At any point during this process Jessica can start making playlists. Eventually Jessica has had enough organizing for one day, saves her graph file using the “Save” option on the file menu, and closes Trax.

5.2 Performance

Overall, it seems clear that Trax has met all of its design goals to one degree or another. It provides a highly interactive means of browsing through a person’s mp3 collection and creating playlists, as well as providing means for editing genre and other information about mp3 files. Whether these means of performing these tasks are better than any other means currently available is not something that can be definitively determined without some form of user testing. Nevertheless, Trax’s genre graph is a unique feature that no other system currently offers, as far as I have been able to determine. This offers a clear distinction from other alternatives, and as such is Trax’s greatest strong point.

The gene graph successfully shows relations amongst genres, with related genres being closer in location on the graph, and more unrelated genres being farther apart. The supergenre classification algorithm successfully classified genres that are generally considered to be high-level genres (such as Rock and Pop) as supergenres by default, while making most other smaller genres into subgenres. Thus the genre graph accurately represents some real-world relations amongst genres for the given dataset, making it clearly a useful tool for conceptualizing and managing genres and their related artists. Though there are some larger theoretical issues with how the genre graph represents genre relations discussed in Section 6.2, overall it seems the genre graph has been successful.

Another area in which Trax performed well was in its attempts at dealing with the potential complexity of the genre graph. The two main means of dealing with this was the supergenre nodes and the genre-centred graph. The supergenre nodes had a definite popout affect against the sea of smaller nodes, and furthermore the ability to filter out all but the supergenres allowed for quick and clear viewing of high-level genres. Furthermore, by allowing users control over the threshold for supergenres, they were given control over the amount of data that they would deal with at one time. The genre-centred graph had a similar affect, in that it greatly reduced the visual clutter and cognitive load required to get a desired piece of information. The genre-centred graph differed from the supergenres in that it allowed users to view both large and small nodes easily, as it greatly reduced the size of the graph displayed down to a manageable subset. Again the user was given control over how much detail they wanted to see.

Other positive aspects were the colour scheme, which clearly indicated which nodes were selected, highlighted or simply normal nodes. Additionally, the colour scheme of light orange and dark red-orange for highlighting, along with blue for selecting, would not cause issues for anyone with red-green colour blindness.

One of the main tasks that was only partially successfully covered was that of playlist making. The actual act of making playlists is very simple and straightforward using Trax, but the fact that you must export the file and then listen to the music in another program is a definite drawback. It had originally been

planned that Trax would be able to play mp3s from the playlist, but that ability had to be scrapped due to time constraints.

Lastly, there are several pieces of functionality are quite important to the overall usability of Trax that was also not added into the system due to Time constraints. Most importantly, there is currently no means of deleting a genre once it is created or removing a genre of a track. This is due to some underlying issues with how the graph is represented that were not fixed in time. Additional unfinished functionality include not being able to reorder tracks on the playlist, and not being able to move a track from one release to another, or from one artist to another. Consequently, while the overall experience of using Trax is quite positive, the program itself cannot be deemed ready to be used by the general public.

6 DISCUSSION

Though Trax clearly succeeded in most of its design goals, there were many interesting issues both positive and negative that were raised by the genre graph and the way it displayed relationships between genres. One of the advantages of the fact that I am both designer of and a potential user of this system is that I have a strong knowledge in both the field of design and the field of use. As such the Trax system provoked some interesting interrelated issues both in visualization and in music and genre.

6.1 Positive Aspects

As discussed in section 5.2, the Trax system performed quite positively overall, achieving most of its design goals fairly clearly. But above and beyond the technical achievements of Trax, there are several more subjective, aesthetic success of the Trax system.

The true success of the Trax system is in its actual visualization of the mp3 collection. Once all the mp3 have been tagged and sorted, browsing through the graph is quite pleasurable. The various levels of detail that the graph can be viewed at and the way one can traverse along a trail of genres by continually double-clicking on nodes is quite pleasant, and a very interesting way of seeing how genres relate to each other. The interaction is quite robust and smooth, aided by Prefuse's inbuilt panning and zooming functionality.

Another positive aspect is the force-directed layout itself. Though the results are not as clean or obvious as they sometimes can be, there is still a definite special relation that occurs between genres that reflects real-world genre relations. For instance, in the data set, the supergenres House and Techno tend to end up quite close to each other, with a collection of other electronic music subgenres floating in an archipelago around them. House and Techno are the two fundamental genre of electronic music, to which almost all the other genres owe at least some fealty to. Other supergenres such as Rock, Pop and Hip Hop are farther away from the pack of electronic music genres, which reflects their distance musically from these genres. Lastly, IDM which tends to be a bit of a magpie of genres, stealing and appropriating elements from a widespread host of different genres, sits somewhere in between the electronic music section and the non-electronic music section. That these layout positions actually reflect real-world facts about the given genres proves that the system really does create an informative visualization of a group of genres that is at least relatively accurate.

6.2 Design Issues

There are two general categories of major design issues that the Trax system still faces. First there were the technical issues, such

as issues of lacking or incomplete functionality or tasks that were not supported as well as they could have been. The second category is the higher-level issues, that do not represent a failure of the system to reach its goal, but rather point to issues with the basic concept behind Trax itself. Though in away these issues are more damning, they are also more thought-provoking, especially to a fan of music. In a way, the fact that Trax provoked so many thought in myself about genres and how they relate to each other proves that it is an interesting and worthwhile representation of genres, though the fact that most of these issues point to shortcomings in the design Trax is troublesome.

6.2.1 Technical Issues

The one main limiting factor of Trax is probably the initial period of time and effort that is needed to make the system usable. As stated above, most mp3 collections do not come in a nicely labelled and tagged set. Some parts may be nicely organized, but inevitably other sections are not. This leads to a rather messy graph at first, and one that does not contain any links. What is required of the user is a good knowledge of different genres of music and how they apply to the user's collection of mp3s—and above all this the desire to put in the time to labelling all these mp3s.

The current technique for labelling genres is reasonably efficient, in that whole artist's oeuvres can be labelled in a particular genre all at once; however, in a collection with 3000 artists, this is still a lot of choices. Unfortunately this is compounded by the fact that genre editing can only be done via a popup menu, selecting an artist, release or track. Originally it had been planned to allow for drag-and-drop adding of artist to genres, as well as creating direct links between genres, but this had to be scrapped due to the lack of support for drag-and-drop in the Prefuse system, and lack of time in general. Graphical editing of genres and the ability to edit groups of artists at a time would greatly help speed the process of getting an initial, rough graph ready.

Another issue is that of the layout of the genre graph, which could be improved in several ways. Currently the force-directed layout algorithm is run anew whenever the graph is created, and thus the layout is different every time the user runs the program. This obvious creates some problems for finding particular nodes. The fact that the supergenres work as anchor points for people finding other genres helps somewhat solve this problem, but it does not totally get rid of the issue. It would be good if users could save a layout once they found one that they liked, and only run the force-directed layout algorithm again if they wished. Even more helpful would be a search function.

Lastly, the current mp3 selection features are fairly limited in their range. Users are only able to sort by artist, than release, than track. This has several problems, firstly in that users might often want to view by releases for such things as compilation albums, as in the current view it is impossible to enqueue all at once an album consisting of many different artists. Also, it would be very useful if users could sort tracks by label as well. Especially in the field of electronic music, the label a track is released on is often as important or even more important than the actual artist, so it would be nice if this was one of the categories that mp3s could be sorted on.

6.2.2 Conceptual Issues

Conceptually, there are probably two main issues with the idea behind Trax. The first is that it assumes that a user's collection of mp3s will be able to be turned into a nice complete interlinked graph which will informatively represent the relations amongst all

of the mp3s' genres. The second is that it further assumes that a graph built up only from a person's mp3 collection is actually what a serious music aficionado would want.

This is all at least somewhat based on the broader issue that it is left to the user to impose a reasonable hierarchy of genres upon their collection. Now, for the target audience of true music obsessive, for whom arguing over micro-genre differentiations is an impassioned pastime, this may actually be a lesser hurdle than may initially be thought. Certainly from my own experience with the Trax system, I rather enjoyed going through and coming up with subgenre-supergenre relations amongst my collection. Though relying on users' music-obsessive nature for the success of this system may limit its widespread applicability, given the many other specialist-oriented visualization systems in existent, it hardly seems out of place in the field. Though almost everyone has an mp3 collection, this system is targeted at only the truly dedicated music fan.

6.2.3 Problems with the Genre Graph

The concept of a genre graph built up from an mp3 file has several problems inherent to it. Likely the main issue is that, because a force-directed layout is used, if there are any genres or groups of genres that are not connected to the larger graph, they will tend to float away off screen and be lost. This is a problem with one of the fundamental assumptions of the Trax system: that all genres are interrelated at some level, in some way. This may not always happen for several reasons. Users may be unwilling to join two genres together, or not having an actual mp3 to represent the connection that they know exists conceptually, or there just may not be any connection between a set of genres that the user knows of.

For instance, I have a node in my graph for Indian music that is currently floating unattached to any other genre of music. Now I could connect this to several other styles of music under the heading of "World Music" but I find that particular genre term objectionable. Unfortunately I don't have any other "natural" connections for the genre to enter into my genre graph, and so it floats off in the distance where it will likely be ignored or forgotten about most of the time. It seems reasonable to assume that this would happen quite frequently in other music collections as well, and so represents a serious issue for Trax.

Another troublesome issue with the representation of the genre graph is that of the supergenre-subgenre relationship. While users are given the ability to increase or decrease the sensitivity of the supergenre threshold, there is a problem with how the threshold is actually calculated. Since the supergenres are calculated by the number of neighbours that it has, genres with more subgenres are more likely to be labelled as supergenres. However, this does not truly capture what one usually thinks of when one thinks of a supergenre-subgenre relationship. A supergenre, generally speaking, is just a genre that encapsulates some overall tendencies of a group of smaller genres. There is nothing in that conception about the *number* of subgenres that is required. Unfortunately, as it is currently calculated, supergenres are rewarded for being easily subdivided into many different parts. Thus some of the nuance of the subgenre-supergenre relation is lost, and the supergenre becomes more of an indication of popularity rather than a classical subgenre relationship.

A good example of the trouble with supergenres in the dataset is IDM, a genre that is classified as one of the 6 supergenres by initial default by Trax. The IDM section of my collection is fairly large, but it is by no means the biggest, and there are several other genres that have more mp3s associated with it. The reason why IDM gets classified as a supergenre is actually due the nature of

the genre itself, as IDM is bit of a magpie of a genre, stealing and appropriating a lot of different styles and tropes other genres. Thus by virtue of its myriad influences IDM is classified as an important supergenre, at the same level as Rock, Hip Hop and Pop, which is definitely misleading.

The alternative solution is to simply make genres that are associated with the most mp3s supergenres. However this has its own attendant problems, in that the results are greatly skewed by the specialization of a user's listening habits. In the dataset used, the most popular genres size-wise would likely be Microhouse and Grime, neither of which would be considered high-level genres; in fact they are quite low-level, with essentially no subgenres of their own. Some combination of both number of neighbours and number of mp3s for a given genre might prove to be a better metric for determining supergenres, but it is unclear exactly as to what weight to give each of the two terms.

6.2.4 Problems Meeting the Users Needs

Unfortunately due to the development schedule of Trax, no user testing was able to be done on the system, so all feedback is based on my own experience using the system. Yet even from my own experience, it became clear that there were some definite issues with the general idea of building up a genre graph exclusively from a user's own mp3 collection.

Essentially, the issue is that the completeness of the graph is totally dependant on the actual mp3 collection of the user. Especially in relation to historical connections in genres, it may be hard to actually represent the real-world links between genres because the user's mp3 collection may lack the critical links between genres. For instance in electronic music, critic Simon Reynolds has posited a continuum of continual genre development running from early Chicago House and Detroit Techno through Acid House and the Rave explosion in England, and from there to Hardcore, Jungle, Drum and Bass, UK Garage and to Grime, the latest incarnation of the so called "Hardcore Continuum" [13]. This is a very well known and widely accepted theory put forth by arguably the genre's most important critic. As a fan of electronic music I considered it rather important that this line of relation exist within the genre map that I created. Unfortunately I could not just "make" this connection exist, because the map was created from my own mp3 collection, which, though large, is by no means definitive. Specifically, the link from Jungle and Drum and Bass to UK Garage was rather hard to make, reflecting my own lack of knowledge in that specific area of music. In the end I hacked in the relation, labelling a few tracks that somewhat fit the bill as both Drum and Bass and UK Garage, just to please my desire to see the relationship exist in my graph.

What this indicates is that for the exacting requirements of a true music obsessive, it may not be enough to build up a graph of genres that exist within a given mp3 collection. The temptation is very strong to create the genre graph that I know exists in the abstract, rather than the one that exists in my own mp3 files. This is a real issue, as such paths of development for different genres are quite frequent in many different genres—Jazz for instance, has a long history starting from the blues and mutating through many iterations before ending in the cul-de-sacs of Avant-Garde and Free Jazz. Consequently, it would seem that the ability to create links and create genres that are not directly reflected in a set of mp3s might be a desirable addition to Trax, though it would also explicitly violate the basic concept that it is built upon: namely building a graph that reflects a person's mp3 collection.

7 FUTURE WORK

There are two levels of future work to be done on Trax. One level can be done immediately as the need for these features is self-evident and the method for their implementation straightforward. The second level requires greater investigation into exactly what should be done and how to do it.

7.1 Short-Term Goals

In section 6.2 several pieces of functionality were outlined that were not completed and time, and thus are keeping Trax from being fully usable. The ability to remove genres, reorder the playlist and move tracks to different releases and artists all need to be implemented before the system is truly useful.

After these basic usability issues, probably the highest priority item is to allow means of graphical editing of genres. This means allowing people to drag-and-drop artists onto genres to add them to that genre, and the ability to draw links directly between genres, thus labeling all mp3 in one genre with the other genre as well. This would greatly aid users in their initial import of mp3s into the system, which remains the biggest usability hurdle for Trax.

After this, the next highest priority would likely be to expand the mp3 selection methods so that they can be sorted by release or label in addition to artists. After this, it would also help the overall usability of the system if it could also play mp3s, in addition to simply exporting playlists of them.

7.2 Longer-Term Goals

As discussed above, there are some deep-seated issues with some of the basic ideas behind Trax, issues that do not have immediately evident answers. A user study would obviously help identify any usability issues that Trax has as well as its overall usefulness in its declared aims. But it would also be an excellent means of determining answers to some of the more troubling issues of the Trax system.

For one, it could be used to determine just what the truly optimal means of calculating supernodes is, by testing different supergenre styles and seeing which one users prefer. Another aspect of the study could also be used to determine if there truly was a desire to create genres and links that do not represent any actual mp3s in a user's collection. It could be determined if the extra usability gained by having these additional nodes and links is outweighed by the confusion of having some nodes that represent actual mp3s, and some that do not.

7.3 Other Work

One of the more interesting suggestions put forth by colleagues was to harness the labeling abilities of Trax to label mp3s according to some other factor other than genre, such as style or mood. Since all the genre information is kept in an xml file rather than in the ID3 tag of an mp3, there is nothing stopping users from using terms other than genres to sort their collection. This opens up a wide range of uses and could lead to some very interesting and informative graphs.

An interesting side effect of this would be that it would also likely lead to having multiple classification schemes for the same set of mp3s. If this was the case, it would then be possible to compare these different classification schemes to determine relationships between them. For instance users might find a high correlation between songs labeled with the "Angry" tone, and songs classified as Heavy Metal—and hopefully some less obvious correlations as well. However, how such a comparison

would be done, and how it would be represented is still very much an open question.

8 CONCLUSION

The Trax system set out to provide a unique, interesting and usable way to organize an mp3 collection and make playlists. It used a node-link graph built up of the genres of an mp3 collection. Mp3s were classified according to multiple genres, and links were created between any two genres that shared an mp3 file. A force-directed layout algorithm was used to layout the nodes in a graph, that a user could then browse. Artists for the associated genres could be viewed and their songs enqueued in a playlist. Multiple means of dealing with the potential complexity of these genre graphs were provided, including separating the nodes into a supergenre-subgenre relationship, filtered views of only the supergenre, and the ability to view a smaller graph of limited depth centered around a given genre.

The results of this system were quite positive, in that it achieved almost all of its initial design goals, and undoubtedly supports the main actions it set out to support. The myriad of different viewing options proved powerful, and the graph itself was pleasurable to interact with, while contemplating how a given track relates to different genres was a interesting intellectual exercise for someone of a music obsessive bent.

There were, however, some issues inherent with the general concept of building up a node link graph of genres directly from a persons mp3 collection. Due to how supergenres are determined, they sometimes do not accurately represent real-world supergenre-subgenre relations. Also, not all genres may be able to be connected—or indeed should be connected—using a persons mp3 collection. This causes islands of unconnected genres. Furthermore, users may wish to have genres and connections between genres that they now exist in the real world, but for which they do not have a representative mp3. In general there are issues that the graph created from a persons individual mp3 collection may not reflect that persons actual beliefs and knowledge about genres and how they relate to each other.

Despite this, Trax is a enjoyable system to work with, and has achieved its main design goals. Furthermore, it has proven quite thought provoking in regards to various issues about musical genres, which is exactly what a music obsessive such as myself enjoys.

REFERERNCES

- [1] Auber, D., Chiricota, Y., Jourdan, F., and Melancon, G., "Multiscale Visualization of Small World Networks" *Proc. InfoVis* 2003.
- [2] Brewster, B. and Broughton, F., *Last Night a Dj Saved My Life: The History of the Disc Jockey*, Grove Press, New York, 2000.
- [3] Brochu, E., "MILQ." *Master's thesis. University of British Columbia*, 2004.
- [4] Eshun, K., *More Brilliant Than the Sun: Adventures in Sonic Fiction* Quartet Books (UK), London, 1999.
- [5] Ganser, E., Koren, Y., and North, S. "Topological Fisheye Views for Visualizing Large Graphs," *IEEE TVCG* 11(4), p 457-468, 2005
- [6] Heer, J., "Vizster: Visualizing Online Social Networks" *InfoSys247, Information Visualization*, Prof. Marti Hearst, School of Information Management and Systems, UC Berkeley, April 2004.
- [7] Heer, J., Card, S.K., and Landay, J.A., "Prefuse: a toolkit for interactive information visualization". In *CHI 2005, Human Factors in Computing Systems*, 2005.
- [8] Herman, I., Melancon, G., Marshall, M.S., "Graph Visualisation in Information Visualisation: a Survey" *IEEE Transactions on Visualization and Computer Graphics*, 6(1), pp. 24-44, 2000

- [9] Lam, H., "Exploratory Browsing in Music Space," *CPSC533C, Information Visualization, Dr. Tamara Munzer*, UBC, 2004
- [10] Leung, Y.K., and Apperleym M.D., "A review and taxonomy of distortion-oriented presentation techniques," *ACM Transactions on Computer-Human Interaction*, Vol. 1, No. 2, June 1994, pp. 126-160.
- [11] Munzer, T., "H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space" *Proc InfoVis 97*.
- [12] Munzer, T., Guimbretiere, F., Tasiran, S., Zhang, L., Zhou, Y. "TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility" *SIGGRAPH 2003*.
- [13] Reynolds, S., *Generation Ecstasy: Into the World of Techno and Rave Culture*, Routledge, New York, 1999.
- [14] Sinha, R., and Sweringen, K., "The Role of Transparency in Recommender Systems" *In CHI 2002 Conference Companion*, 2002.
- [15] Van Ham, F., and van Wijk, J. "Interactive Visualization of Small World Graphs" *Proc. InfoVis 2005*