

COMBINING METADATA, INFERRED SIMILARITY OF CONTENT,  
AND HUMAN INTERPRETATION FOR MANAGING AND LISTENING TO  
MUSIC COLLECTIONS

A Dissertation

by

KONSTANTINOS A. MEINTANIS

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2010

Major Subject: Computer Science

Combining Metadata, Inferred Similarity of Content, and Human Interpretation for  
Managing and Listening to Music Collections  
Copyright 2010 Konstantinos A. Meintanis

COMBINING METADATA, INFERRED SIMILARITY OF CONTENT,  
AND HUMAN INTERPRETATION FOR MANAGING AND LISTENING TO  
MUSIC COLLECTIONS

A Dissertation

by

KONSTANTINOS A. MEINTANIS

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Frank M. Shipman
Committee Members,	Richard Furuta
	John Leggett
	Jimmie Killingsworth
Head of Department,	Valerie E. Taylor

August 2010

Major Subject: Computer Science

## ABSTRACT

Combining Metadata, Inferred Similarity of Content, and Human Interpretation for  
Managing and Listening to Music Collections. (August 2010)

Konstantinos A. Meintanis, B.S., Athens University of Economics and Business

Chair of Advisory Committee: Dr. Frank M. Shipman

Music services, media players and managers provide support for content classification and access based on filtering metadata values, statistics of access and user ratings. This approach fails to capture characteristics of mood and personal history that are often the deciding factors when creating personal playlists and collections in music. This dissertation work presents MusicWiz, a music management environment that combines traditional metadata with spatial hypertext-based expression and automatically extracted characteristics of music to generate personalized associations among songs. MusicWiz's similarity inference engine combines the personal expression in the workspace with assessments of similarity based on the artists, other metadata, lyrics and the audio signal to make suggestions and to generate playlists. An evaluation of MusicWiz with and without the workspace and suggestion capabilities showed significant differences for organizing and playlist creation tasks. The workspace features were more valuable for organizing tasks, while the suggestion features had more value for playlist creation activities.

DEDICATION

To my parents

## ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Frank Shipman, and my committee members, Dr. Richard Furuta, Dr. John Leggett, and Dr. Jimmie Killingsworth, for their guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to my mother, father, brothers and sister for their encouragement patience and love.

## TABLE OF CONTENTS

	Page
ABSTRACT .....	iii
DEDICATION .....	iv
ACKNOWLEDGEMENTS .....	v
TABLE OF CONTENTS .....	vi
LIST OF FIGURES.....	viii
LIST OF TABLES .....	x
1. INTRODUCTION.....	1
2. PROBLEM .....	4
3. RELATED WORK .....	7
3.1 Personal Music Collections .....	7
3.2 Music Digital Libraries .....	13
3.3 Related Work Summary .....	14
4. EXPRESSION OF PERSONAL INTERPRETATIONS OF MUSIC COLLECTIONS IN SPATIAL HYPERTEXT.....	16
4.1 Spatial Hypertext.....	16
4.2 Study Design .....	18
4.3 Study Results.....	19
4.4 Implications for System Design .....	30
4.5 Preliminary Study Summary .....	33
5. MUSICWIZ DESIGN .....	35
5.1 Interface and Functionality.....	37
5.2 Inference Engine .....	48

	Page
6. MUSICWIZ MUSIC SUMMARIZATION .....	67
6.1 Introduction .....	67
6.2 Current Research .....	68
6.3 Algorithms.....	71
6.4 Evaluation Design .....	76
6.5 Evaluation Results.....	77
6.6 Discussion .....	82
7. MUSICWIZ EVALUATION.....	84
7.1 Evaluation Design .....	84
7.2 Evaluation Results.....	88
7.3 Discussion .....	98
8. CONCLUSION AND FUTURE WORK.....	106
REFERENCES .....	111
VITA .....	115



## LIST OF FIGURES

FIGURE		Page
1	The Standard List View of the Music Library in iTunes .....	8
2	Organization Using Categories, Subcategories and Labels .....	17
3	Single-Level Organization Using Collections, Color, and Border Width..	20
4	Organization and Playlist Creation Based on Preference.....	21
5	Organization Using Spatial Layout to Indicate Preference .....	23
6	Visual Expression in the Workspace.....	27
7	Example Playlist.....	28
8	MusicWiz's Architecture .....	35
9	MusicWiz's Interface Combines a Tree View, a Workspace, and an Area for Search Results and Related Music.....	36
10	MusicWiz's Visual Attributes Controls .....	37
11	Examples of Song and Plain Objects inside Collections in Spatial Formation .....	40
12	MusicWiz's Tree View, Similar View (a) and Search View (b) .....	41
13	MusicWiz's Configuration Dialog .....	42
14	MusicWiz's Search Menu .....	43
15	MusicWiz's Playback Controls .....	43
16	MusicWiz's Playlist Pane.....	44
17	MusicWiz's Playlist Properties Menu.....	45
18	FFT-based Approach for Beat Extraction of C. Isaak's <i>Wicked Game</i> .....	52

FIGURE		Page
19	DWT-based Approach for Beat Extraction of C. Isaak - <i>Wicked Game</i> ....	53
20	DWT-based Approach for Beat Extraction of Dire Straits - <i>Sultans of Swing</i> .....	55
21	Brightness Levels over Time for two C. Santana's Songs.....	57
22	Three Levels of Accuracy / Speed for Pitch Extraction.....	59
23	Pitch Analysis of Four Songs in the Best Quality Option.....	60
24	Example of the Structures recognized by MusicWiz's Spatial Parser.....	63
25	Important Parts for Recalling Music .....	77
26	Important Parts for Familiarizing with Music.....	78
27	Users' Algorithm Choice .....	78
28	Evaluation of Summaries' Performance .....	80
29	Participants' Last Time of Listening to the Song.....	82
30	Music Education Levels and Collection Size.....	84
31	Music Collection Structure – Organization and Listening Habits .....	86
32	The Average Completion Times of the Participants in the Three Tasks....	88
33	Organization Based on Music Genre and Dynamics (Group Four).....	90
34	Organization Based on Music Knowledge, Concept of Listening, and Music Dynamics (Group Three).....	91
35	Genre-based Classification in Windows Folders (Group Two).....	92
36	Preference and Concept-based Playlist Creation (Group Four).....	95
37	Concept and Event-based Playlist Creation (Group Two).....	96

FIGURE	Page
38 Playlist Creation by Example (Group Three).....	97

## LIST OF TABLES

TABLE		Page
1	The Textual Descriptors Used for Collections and Playlists by Study Participants.....	24
2	Pair-wise Comparison of the Four Algorithms .....	79
3	Algorithm Selection and Familiarity with Music.....	81
4	Participants' Preference on Genre.....	85
5	MusicWiz's Configurations for Study Groups.....	87
6	Average of Seven-point Likert-scale Ratings for Playlist Creation - Higher Values Are More Positive .....	94
7	Collection and Playlist Labels Created at Task One and Two.....	99
8	Number of Collections per Group and Type in Task One .....	101

## 1. INTRODUCTION

For the majority of people, the management of their personal music collections means associating and classifying songs according to their explicit attributes. Metadata values like the artist, the composer and the genre of music are used extensively in determining the classification scheme and its components. Undoubtedly, a taxonomy based on explicit attributes has many advantages. Files can be easily and consistently classified, searched and retrieved while the classification schemes “make sense” to almost everybody since they are based on well-defined criteria. Accordingly, the applications for processing those schemes provide accurate access and filtering of the music as they are designed based on the same principle of a song as a file identifiable by well-defined attributes.

The common metadata fields attached to music are valuable for providing context-free information about the music – the artist of a recording does not change between playbacks – but are not necessarily the music characteristics that express what users really seek. Searching songs by how the music makes you feel or how the music sounds is currently very difficult. How can users pick music that they find happy, energizing, or calming? What about music that reminds them of high school, or of college, or of particular family members or friends? While they can add metadata fields, they rarely do so [Shipman and Marshall 1999] because the potential value is outweighed by the overhead of the expression, especially when that expression involves

---

This dissertation follows the style of ACM Transactions on Information Systems.

interpretation that is likely to change over time, such as the feelings and memories triggered by listening to the music. Retrieving songs based on previous explicit feedback (e.g. ratings) and access statistics can reliably detect music of preference but not necessarily music pertinent to a specific mood or feeling.

Describing taste and thoughts, especially complex ones, using explicit means is neither sufficient nor efficient. Does this imply that explicitness in managing music is unnecessary and that replacing existing systems with technologies supporting implicit expression will solve the problem? The answer is probably no. A collection taxonomy based on implicit associations is far more sensitive to changes versus an organization where membership is decided based on well-defined and consistent criteria. Music understanding, perception, and mood are volatile factors that can differ not only from person to person but also from time to time. An organization relying exclusively on those factors can be so dynamic that locating specific sources can be far more complicated than filtering metadata values or statistics.

The problem of managing and using personal music collections is neither trivial nor simple. On one hand, systems need the consistency, compatibility, accuracy, and formality of explicit expression. On the other hand, they also need to support less restrictive / more abstract models of implicit expression. Current multimedia applications are quite efficient when working with data for which there is an explicit description. However, this dissertation is based on the belief that they can serve user needs even better by taking into consideration implicit expression as well. This combination is embodied in MusicWiz: an environment that encourages / supports

associating songs based on the personal feelings and memories of the user, and techniques for identifying, retrieving and navigating related music using a combination of implicit and explicit criteria.

Section 2 describes the problem of music management and use and the potential for freeform visual expression, as found in spatial hypertext, to facilitate these activities. Section 3 presents an overview of related work in the areas of music management, music similarity assessment, visualizations for presenting music collections, and digital libraries of music. A preliminary study exploring the potential and issues associated with using spatial hypertext for managing music collections is presented in Section 4. The results of this study led to the design of MusicWiz that is found in Section 5. MusicWiz incorporates traditional metadata access to collections with visual expression and a similarity engine that combines a wide variety of forms of similarity assessment to make suggestions and fill out playlists for users. Providing access to the contents of a music collection means playing/presenting songs so that users can determine whether a song fits their current needs. Section 6 presents a novel approach to generating music summaries to be used as previews of songs in MusicWiz. The evaluation of MusicWiz, in particular the inclusion of a visual workspace for organizing the elements of a music collection and the inclusion of suggestions, is found in Section 7. Section 8 presents conclusions and areas for future work

## 2. PROBLEM

In today's commercial products, the generation of playlists is a matter of finding songs with overlapping metadata values or statistics. Value matching ensures that the resulting collections have some coherence. However, taking into consideration metadata similarity is just one form of relatedness and has many limitations. The problem has two dimensions. First, the explicit information that is used to characterize a resource (metadata and statistics) is usually fixed in content, general and very concise for efficiency and consistency reasons. As a result, its expressive power is insufficient for depicting complicated or domain specific concepts. How can we describe (in terms of ID3 tags) for example all the four-voice fugues in minor scale by J. S. Bach without specifying their exact title and number? Second, the filtering mechanism works well only when there is a strict and formal description of the searching pattern. Hence, identifying concepts that are too abstract or very specific for formal description is difficult and usually fails as the results tend to be inaccurate, incomplete (subsets of the correct answers) or redundant. In the previous example, the metadata description that probably fits best the given concept is classical music (genre) by J. S. Bach (composer) from his collection Preludes & Fugues or Toccata & Fugue or Passacaglia & Fugue (supposing that there is such an album name). Obviously, a search using those values will return not only the fugues by J. S. Bach written in minor scales but also his fugues in major scales, their preludes, a toccata, and a passacaglia.



The general hypothesis behind this research proposal is that, if we use a combination of explicit and implicit information to compare music, we can determine songs' relatedness in a more reliable and comprehensive way. Implicit information in music can be anything non-explicitly assigned to a song that can be derived by analyzing its actual content (e.g. audio signal attributes and lyrics similarity) and its membership in a collection or category (e.g. associations with other files that may reside in the same playlist). Although it can be difficult and costly to extract it, implicit information is not constrained by the limitations of formal representation and hence it can be especially rich and domain specific. It can contain details about how a song is harmonically and dynamically structured (e.g. music identification based on frequency range and loudness evolution in time), personal preferences (e.g. what are the music attributes/style of the songs the user likes), and collection management practices (e.g. what are the music attributes/style of the songs belonging to the same collection or what differentiates songs of different collections). The author believes that by carefully combining the efficiency / consistency of the explicit attributes with the descriptive power of the implicit expression it is possible to retrieve songs that create the "fit well together" feeling people look for when they create playlists manually.

A challenge for building an environment that includes more implicit information about the user's perspective on a piece of music is that it is constantly changing. Consider a hypothetical user Susan. Susan does not like a piece of music the first time she hears it but it grows on her as she hears it more. Eventually, this song becomes associated with the people and things in Susan's life when the song was popular and/or

she was listening to it. Depending on Susan's feelings about these people and activities, her opinion of the music will continue to evolve.

To build a system that allows users to convey such changing assessments requires that any necessary user expression is very light weight – that is, it takes little time and it is easily modified. Expression through the traditional metadata application of attributes and values does not meet this goal. Tags, which are just text attributes without values, are simpler to apply but still require the user to express their reaction to a piece of music in words, something that may not be easy. Finding a medium of expression that removes this requirement is likely to reduce the effort of expression.

Spatial hypertext systems were designed to support the rapid expression of categories and associations between documents through the application of visual features (e.g. color or border width) to document objects and through the placement of objects in visual proximity or structures. Thus, one aspect of this dissertation explores the effects of adding visual expression as part of the management of music collections.

Once personal expression concerning the elements of a music collection has occurred, the music management environment can use that expression to make suggestions to the user. Because the visual expression is meant to capture aspects of the music not encoded in traditional metadata, ratings, playback statistics, or the audio content itself, the environment must combine these different forms of information when deciding on suggestions. Thus, a second aspect of this dissertation is the multi-faceted calculation of similarity between songs.

### 3. RELATED WORK

Previous research on providing access and suggestions to personal music collections and previous research on digital libraries is related to the current dissertation.

#### 3.1 Personal Music Collections

The related work in personal music collection management falls into two main categories: systems that rely exclusively on explicit attributes (like metadata and ratings) to organize and retrieve music and systems that use implicit information or a combination of explicit and implicit information.

##### *3.1.1 Music Access Based on Explicit Attributes*

Most of the commercial products for playing and managing music use explicit attributes. Popular examples include media players like the iTunes (Figure 1), Windows Media Player, the QuickTime Player, and the Real Player as well as media managers like the Media Monkey, the Media Catalog Studio, and the Songs-DB. The former systems focus on supporting access to and playback of media files while the latter put more emphasis on tasks related to the organization of music collections. In both cases, systems support manipulation via metadata tags and album information, statistics of recency and frequency of access, and user preference in the form of ratings. Hierarchical views of the songs as they are stored in the file system facilitate the manual search of the music collection. As applications oriented to support collection management, media managers provide additional functionality for the creation, revision and online lookup of tags, as well as the division of collections into sub-collections, and the restructuring of



Figure 1. The Standard List View of the Music Library in iTunes

collections into logical hierarchies based on metadata values. Access to a music collection in these systems is through a combination of search and browsing. The user can specify the values (absolute or range) of the explicit attributes of interest and the application returns the matching songs. To browse a collection, the user selects and navigates a hierarchic view of the collection. Alternate views use different metadata values to group music into different hierarchies. To improve suggestions based on music similarity, Pandora internet radio ([www.pandora.com](http://www.pandora.com)) uses a large number of music experts to classify and associate songs according to a pool of 400 musical attributes.

Given a song, an artist or a keyword as seed, the system searches for overlaps in the human assigned attributes and returns a playlist with the best matches. Results can then be filtered further based on previously provided user feedback.

Visualizations based on explicit attributes provide another form of access to a collection. In one example, van Gulik and colleagues [van Gulik et al. 2004] present visualizations of clustered music to aid access on small screen devices. Their system clusters songs based on their metadata and mood provided by the MoodLogic music meta-database (<http://www.moodlogic.com>). In addition, it provides an *artist map* overview of the entire collection and a view of artist similarity. To assess artist similarity, the system computes the feature vectors of the songs for each artist. Based on these vectors, it generates the histogram that corresponds to the style of each artist. The comparison of the histograms gives the “distance” between the artists.

### 3.1.2 *Music Access Including Implicit Attributes*

In an effort to escape from the limitations of using metadata to describe custom music concepts and the unwillingness of users to provide explicit feedback, there is considerable research into extracting and using implicit cues for associating music.

Instead of using metadata, many systems use assessments of music similarity based on sound and melody features. Liu, Lu and Zhang [Liu et al. 2003] extract the intensity, timbre and rhythm of songs and combine them to detect mood. Logan and Salomon [Logan and Salomon 2001] use the Earth Movers Distance (EMD) to compare song signatures generated by analyzing the Mel-Frequency Cepstral Coefficients (MFCCs), the loudness, and other dynamic characteristics of the music signal. Similarly,

Aucouturier and Pachet [Aucouturier and Pachet 2002] measure distance by calculating the matching likelihood of samples from the Gaussian Mixture Model of the songs. Hoashi, Zeitler and Inoue [Hoashi et al. 2002] have developed a content-based music retrieval method that uses Foote's tree-structured vector quantization algorithm TreeQ [Foote 1997]. The algorithm first assigns audio samples from a training set into the bins (leaves) of a quantization tree. Training sets exist for different attributes of music (e.g. genre, artist) and the assignment is based on the spectral representation of the samples. The algorithm applies vector similarity measures in order to find songs that have relative frequencies similar to those of the samples in the quantization tree. To improve retrieval performance, a relevance feedback mechanism refines the category vectors generated by the TreeQ method. Supporting access based on similarities between audio signals avoids the need for metadata but relies on signal processing techniques to match user assessments.

Rather than solely relying on explicit metadata or signal processing, another set of systems support access to music by making inferences based on the activity and expression of groups of users for alternate purposes. Instead of using human experts to identify relevant music like in Pandora, Last.fm (<http://www.last.fm>), another internet radio but also a music community website, utilizes the power of the collaborative filtering (CF) to create one of the most popular music recommendation systems today. Last.fm users have a detailed taste profile that is constantly updated according to their music selections and feedback (they can "love", "skip" or "ban" a song) on the streamed radio stations, their personal computer or their portable music devices. Profiles are

modifiable and users can manually enrich for example their “loved” tracks or remove songs from the list with the banned ones. In the Last.fm network users can have friends, create or join groups and participate in events. Collaborative filtering algorithms analyze their social activity and membership and generate recommendations for artists that appear in profiles sharing similar musical tastes. Users can also recommend artists, songs or albums directly to others (individuals or groups) while they can listen to “recommendation radio” featuring all the artists that have been recommended to them. The Genius application in the iTunes media player by Apple Inc. is another popular commercial product that uses collaborative filtering for recommendation of similar music. Recommendations are based on the music that is on the user’s personal collection, her purchases from the iTunes Store and what other people with similar taste have listen to and bought in the past. Similarly, Zadel’s and Fujinaga’s Music Information Retrieval (MIR) web service [Zadel and Fujinara 2004] takes an artist as a seed and assesses the similarity of other artists by measuring their co-occurrence in the Amazon Listmania! Database. van Breemen and Bartneck’s Music Gathering Application [van Breemen and Bartneck 2003] adopts a similar web-based architecture where agents download songs from the OpenNap servers (<http://opennap.sourceforge.net/>) based on user’s existing collections and behavior as well as what is popular according to several music websites. Crossen and colleagues [Crossen et al. 2002] collaborative recommendation system (Flytrap) extracts metadata about the artist and the genre of the songs that the user listens to. The system determines similarity among artists through the use of a hand-built semantic network of interrelated

genres. A voting mechanism decides the applicability of each song according to similarity of the genre and whether the artist has been selected in the past. All of these techniques use human activity that occurs for one purpose to support another. Li and colleagues [Li et al. 2004] have developed a collaborative music recommender system (CMRS) that selects music using collaborative filtering as well as content-matching algorithms to extract and match the timbral texture and rhythmic pattern of the songs.

Collaborative filtering is not the only approach for utilizing user feedback in associating and suggesting music. The “interactive web-Radio” *Musicoverly* (<http://musicoverly.com/>) provides implicit cues for associating and navigating music. In its “mood pad”, users can quantify how “Dark” and “Energetic” the song selections to be by mouse clicking on a continuous, 2-D space. The X-dimension of that space is mapped to how “Positive” the music sounds while the Y-dimension to how “Calm”. In the “tempo pad”, which is an alternative view of the same space, users can customize similarly the “Tempo” and “Dance” factor of music. Selections can be refined by filtering the music based on attributes like genre or release year. Towards the direction of providing a relaxed way of managing and browsing music collections, Y. Chen and A. Burtz have developed *MusicSim* [Chen and Butz 2009]. *MusicSim* provides a “graph view” to display the songs as 2-D objects clustered according to their content-based similarity and previous user feedback. Songs are positioned relative to the cluster center according to their similarity to the centroid and other neighboring songs. The main difference of the proposed interface is that it is not another collection visualization for music navigation and exploration like that found in the *Islands of Music* [Pampalk et al.



2002], the *Globe of Music* [Leitich and Topf 2007], the *nepTune* interface [Knees et al. 2007], the *MusicBox* [Lillie 2008] or the *PlaySOM* [Neumayer et al. 2005]. In MusicSim, the location of the songs is not fixed and users can reposition them (within the same cluster or to another one) according to their own perception of similarity and hence influencing system's assessments of related music. Users can also pan and zoom the visualization and apply filters based on genre. Goto and Goto [Goto and Goto 2005] propose a highly interactive graphical environment that supports the discovery of similar and unfamiliar music. In *Musiccream*, songs, grouped and color-coded based on the similarity of their mood, stream down, one after the other, from taps on the top of the screen. Users can select falling songs for listening or use the “similarity-based sticking function” to “stick” music they want to listen to the same playlist. The songs comprising a playlist can be rearranged while multiple playlists situated on the screen can be ordered for continuous playback.

### 3.2 Music Digital Libraries

In music digital libraries, browsing and retrieving related resources can be quite demanding considering the volume and nature of the information at hand. To improve the efficiency and accuracy of access, many interfaces provide ways of searching the music content other than the traditional metadata filtering.

In the VocalSearch music search engine [Pardo et al. 2008], users can query the database not only by text-based lyrics but also by singing the melody or music notation. The *Son of Blinkee* (SOB) system and its underlying Networked Environment for Music Analysis (NEMA) [Downie et al. 2008] provide a visualization of (machine-generated)

audio-based classifications (e.g. genre, mood, artist, etc.) that is synchronized to the music playback. As the music progresses, users can see in real time, for instance, the evolution of the different moods or genres involved and hence realize the overall style of music or the association of the various classifications. Hanna and colleagues [Hanna et al. 2009] propose a retrieval system that is based on the similarity in the chord progressions. Chord progression comparison is also used by Kuo and Shan [Kuo and Shan 2004] in combination with instrument, volume and highest pitch information for music classification based on the melody style. Tsai and Wang [Tsai and Wang 2005] propose a music digital library architecture where songs are classified and accessed based on vocal-related information and more specifically the voice characteristics of their singers. Recognizing the importance of associating user-generated opinions to music objects, Downie and Hu [Downie and Hu 2006] analyze online music reviews to find the kind of terms people use to comment negatively or positively. Bischoff and colleagues [Bischoff et al. 2009] have developed algorithms for creating mood (opinion) and theme (occasion) classifiers as well as genre predictors based on user annotations (tags extracted from Last.fm) and lyrics.

### 3.3 Related Work Summary

The variety of prior research and applications supporting music management and selection shows the use of a wide range of information about songs. Environments emphasizing collection management and access tend to provide views on traditional metadata or through automatically computed visualizations of the collection. Applications aimed at selection emphasize similarity assessment based on metadata,

human-coded or automatically extracted musical features, or the co-occurrence of music in playlists or in the favorites of individuals. This dissertation builds on these approaches by including visual expression in the management and use of music collections.

#### 4. EXPRESSION OF PERSONAL INTERPRETATIONS OF MUSIC COLLECTIONS IN SPATIAL HYPERTEXT

Before beginning the design of a new system, a preliminary study was performed to see how low-cost expression influences personal music organization. Spatial hypertext environments are designed to reduce the overhead of user expression for ambiguous and difficult to describe concepts. Thus, it is a good medium for understanding what characteristics of music people want to express that they currently do not and the roles of such expression.

##### 4.1 Spatial Hypertext

Spatial hypertext emerged from node-and-link and map-based hypertext in order to better support the evolving and emergent interpretations that often occur during the early stages of information analysis tasks. A spatial hypertext consists of a set of *information objects* with visual attributes (e.g., color, border width) and spatial layout (e.g., lists, piles) to indicate relations between information entities [Marshall and Shipman 1995].

Due to its availability and access to the source code in case there needed to be some modifications, the initial study had participants use the Visual Knowledge Builder (VKB) [Shipman et al. 2001c], a general-purpose spatial hypertext system where information is placed into a hierarchy of two-dimensional visual workspaces called *collections*.

The barrier of expression in spatial hypertext is reduced relative to traditional hypertext because the assignment of visual attributes and arrangement of objects requires less effort than creating explicit links and relations between objects. By providing a wide range of modifiable visual attributes and the ability to organize materials in space, users can express a variety of relations and their strengths without having to verbally express the meaning and degree of relations. Figure 2 shows the music organization in VKB created by a study participant. VKB displays the title and artist of the audio file in the object for each song. When the cursor lingers over the border of an object, additional

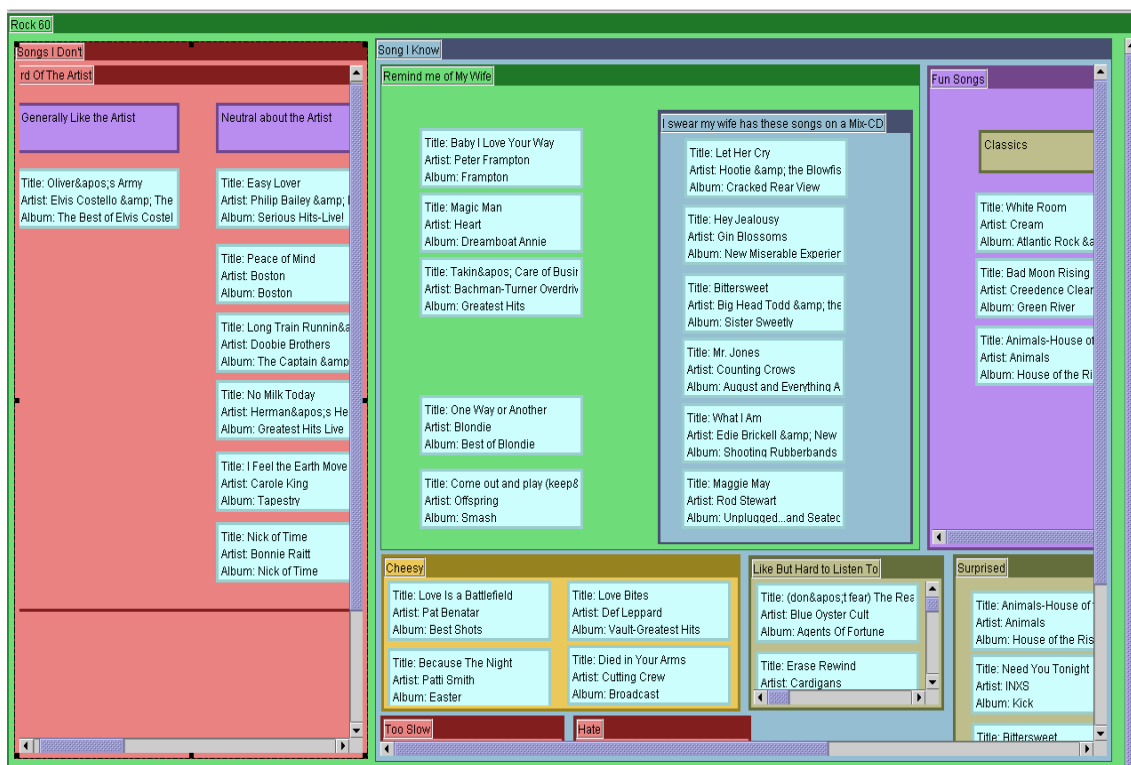


Figure 2. Organization Using Categories, Subcategories and Labels

metadata is shown in a popup. The color and border width variations are the result of the user's expression.

VKB plays the audio file while the mouse cursor lingers over an object. This is an auditory form of progressive disclosure similar to providing metadata, snippets of content or thumbnail images of textual or image content in a popup. In the study, due to limitations with VKB software, audio playback on mouse-over was limited to the first 10 seconds of a song. Subjects could still listen to the whole music file in Windows Media Player by double-clicking on the object.

#### 4.2 Study Design

The study was conducted in the Center for the Study of Digital Libraries at Texas A&M University. Twelve graduate students, age 24 to 38, were recruited to take part in the study including 10 men and 2 women. There was no compensation. The majority (75%) of the participants had previously used VKB for other tasks, reducing the impact of software novelty on results. Regardless of prior experience, participants were trained in the use of VKB prior to the study task.

Collections of 100 pre-selected songs were created for four music genres (rock, dance, lounge, and classical). The participants were asked to select a genre and then given 60 minutes to organize the songs for that genre. Participants were encouraged, but not forced, to "think out of the box" of the traditional metadata classification and to create collections based on their own interpretation of the music. After their organization was complete, participants were asked to create three playlists for activities or events of their own choosing. Subjects were allowed up to 30 minutes to create the playlists.

Demographic data about the participants was collected via a pre-task questionnaire. The organizational process and results were recorded via monitoring code built into VKB, screen capture software, and the resulting VKB files. Post-task questionnaires and semi-structured interviews were used to gather information about the participants' perceptions of the task, tool, and experience as well as their strategies and practices in organization and expression.

### 4.3 Study Results

All participants completed the first task of organizing the 100 songs. Eleven of the twelve participants completed the second task of creating three playlists, with one participant not having the time to continue through this task. The results below are organized into data concerning participants' use of and satisfaction with existing software for organizing and listening to music, their organization of songs into collections and playlists in VKB, and their post-task assessments concerning the task, VKB, and what features they would want in a system to support this task.

#### 4.3.1 *Experience with Digital Music Collections*

The pre-task questionnaire asked about participants' experience in using applications for managing songs and generating playlists. All participants had previous experience in organizing songs. 67% (8 of 12) have a collection with more than 200 songs and 67% spend at least 15 minutes organizing their music every week. Most participants spend a significant amount of time listening to their collection. 67% spend more than 30 minutes during each sitting, and 50% listen to or organize their songs more than 5 times each week. Only one subject was satisfied with the playlist creation

techniques found in most commercial and freeware software where selection and ordering of songs is based on filtering metadata and usage statistics. 83% (10 of 12) replied that they create their playlists manually by browsing their collection and dragging-and-dropping songs into their players.

#### 4.3.2 Organization and Expression

Participants selected only two genres, with four participants selecting rock and eight participants selecting classical. Because the music was pre-selected, participants were confronted with both songs they knew and songs they did not know.

Figure 2 shows part of a finished workspace. The participant divided the songs

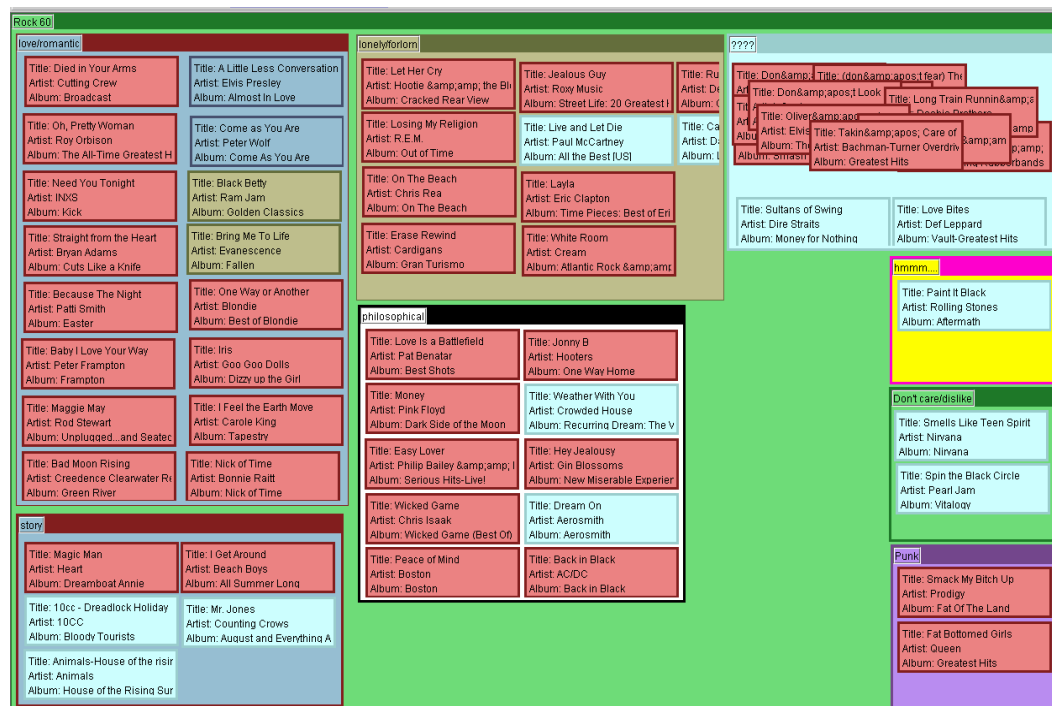


Figure 3. Single-Level Organization Using Collections, Color, and Border Width



into those he knew and those he did not. The unknown songs were organized based on the participant's opinion about the artist (“generally like the artist”, “neutral about the artist”). The songs he knew were grouped based on personal assessments of the music (“like but hard to listen to”, “cheesy”, “hate”, “fun songs”, and “too slow”) and associations the music had for the participant (“remind me of my wife”). Some of these categories had further subcategories such as the “I swear my wife has these songs on a mix-CD” under “remind me of my wife” and “classics” under “fun songs”. This participant's workspace shows a greater degree of structure and interpretation than the workspaces created by most of the participants.

Figure 3 shows a one-level categorization of another participant with the explicit

The screenshot displays a music workspace titled "Rock 60" with several sections:

- Stuff I would listen to:** A grid of song cards categorized by frequency:
  - Anytime anywhere (Green):** Hey Jealousy (Gin Blossoms), Bring Me To Life (Evanescence), Iris (Goo Goo Dolls), Mr. Jones (Counting Crows), Don &apos;apos; (No Doubt), Sweet Child O &apos; (Guns N &apos;), One Way or Another (Blondie), Dream On (Aerosmith), Died in Your Arms (Cutting Crew), Losing My Religion (R.E.M.).
  - Everyso often (Purple):** Maggie May (Rod Stewart), Oh, Pretty Woman (Roy Orbison), Easy Lover (Philip Bailey &apos;), Let Her Cry (Hootie &apos;), Straight from the Heart (Bryan Adams), Baby I Love Your Way (Peter Frampton).
  - Occasionally (Yellow):** (don &apos; (Agents Of Fortune), Need You Tonight (INXS), Love Is a Battlefield (Pat Benatar), I Get Around (Beach Boys), I Feel the Earth Move (Carole King), Takin &apos; (Carmichael &apos;), Bittersweet (Big Head Todd &apos;), Wicked Game (Chris Isaak), Nick of Time (Bonnie Raitt), Peace of Mind (Boston).
- Stuff I probably don't want to appear often, if at all (Blue):** No Milk Today (Herman &apos;), Don &apos; (No Doubt).
- Playlist 1 - Things I can listen to over and over again (Red):** Hey Jealousy, Bring Me To Life, Iris, Mr. Jones, Don &apos;, Sweet Child O &apos;, One Way or Another, Dream On, Died in Your Arms, Losing My Religion, What &apos; (Four Non Blondes).
- Playlist 2 - All decent, interspaced (Yellow):** (don &apos;, Need You Tonight, Easy Lover, Iris, Love Is a Battlefield, I Get Around, Baby I Love Your Way, Wicked Game, Nick of Time, Peace of Mind.
- Playlist 3 - Relative (Blue):** Need You Tonight, Oh, Pretty Woman, Bring Me To Life, Love Is a Battlefield, Don &apos;, Sweet Child O &apos;, Straight from the Heart, Baby I Love Your Way, I Get Around, One Way or Another, Peace of Mind.

Figure 4. Organization and Playlist Creation Based on Preference

categories related to the lyrics, themes and mood of songs. There are collections for “story” songs, “philosophical” songs, “love/romantic” songs, and “lonely/forlorn” songs. In addition, there are categories for style of music (“punk”), for disliked songs, and for songs that did not fit into categories he had already created. This participant used color and border width to indicate features of the music beyond his labeled categories. Figure 4 shows a categorization based on personal preference. At the high level, the music has been split between songs the participant would like to listen to and songs he “doesn’t want to appear often, if at all”. In a second level of refinement, the music assigned to the former collection has been classified further as “anytime, anywhere”, “occasionally” and “every so often”. On the right part of the workspace, the participant has created his three playlists with descriptors “things I can listen to over and over”, “relatively random”, “all decent, interspaced”. Figure 5 shows a categorization based on a mix of music preference and music content. “Favorite” songs have been placed on the top of the workspace followed by “Peaceful”, “Delightful” and “Normal” songs. Music that the participant does not particularly like has been “hidden” in a less visible spot at the bottom of the workspace.

Favorite			
Title: Ludwig Van Beethoven Sonata No.8 In Artist: Alfred Brendel Album: Beethoven: Favourite Piano Sonatas	Title: Frederic Chopin: Ballade For Piano #1 Artist: Krystian Zimerman Album: Chopin: Four Ballades, Barcarolle, F	Title: Jean Sibelius - Finlandia, Op.26 Artist: Various Orchestras Album: Sibelius - Violin Concerto In D M. Op	Title: Fernando Sor - Introduction, Theme, & Artist: Segovia, Andres Album: The Legendary Segovia
Title: Johannes Brahms Hungarian Dance 1 Artist: Claudio Abbado, Vienna Philharmonic Album: Hungarian Dances	Title: Franz Schubert - Work(s) Ständchen Artist: Budapest Strings Album: Masters of Classical Music Vol 9: Sc	Title: Isaac Albeniz Suite española: No. 3: Si Artist: Enrique Báltz, State of Mexico Sympho Album: Albéniz: Iberia & Suite española	Title: Claude Debussy Suite bergamasque Artist: Zoltan Kocsis Album: Debussy - Suite bergamasque, Imax
Title: W.A.Mozart Symphony No. 40 in G min Artist: Böhm, Karl - Berlin Philharmonic Orct Album: Mozart: Symphonies Nos. 40 and 41			
Peaceful			
Title: Georges Bizet - Carmen - Suite: Interme Artist: Bastille Opera Orchestra/Myung-Whur Album: Nightmoods: Twilight Hour	Title: Aaron Copland - Appalachian Spring, c Artist: Hugh Wolff Album: Night Tracks	Title: Edward Elgar - Enigma Variations Op. Artist: Giuseppe Sinopoli/The Philharmonia Album: Nightmoods: Twilight Hour	Title: Vincenzo Bellini - Casta Diva Artist: Maria Callas Album: Maria Callas
Title: Edvard Grieg - Lyric Pieces for piano, E Artist: Cyprien Katsaris Album: Night Tracks	Title: Robert Schumann Piano Concerto in A Artist: Dinu Lipatti Album: Great Pianists Of The Century Vol.3	Title: Maurice Ravel - Ma Mère (à propos; Oye: F Artist: Berlin Philharmonic Orchestra/Pierre I Album: Nightmoods: Twilight Hour	Title: Sergei Rachmaninov Etude-tableau in Artist: Nikolai Lugansky Album: Great Pianists Of The Century Vol.5
Delightful			
Title: Astor Piazzolla - Tango Suite Fugata Artist: Yo-Yo Ma, Kathryn Stott Album: Soul of the Tango: The Music of Asto	Title: Enrique Granados Danza Espanola Nr. Album: Julian Bream Plays Granados &nc Track: 3	Title: Nikolaos Skalkottas - Peloponnesian D Artist: Nikolaos Skalkottas Album: Joseph James: Concerto For 3 Bouz	Title: Dimitri Shostakovich Sonata No.2, Op.1 Artist: Yegorov &nc Album: Great Melodies of the Classics
Title: Johann Jacob Froberger - Gigue, for l Artist: Segovia, Andres Album: The Legendary Segovia	Title: Nikolai Rimsky-Korsakov Flight of the E Artist: Vladimir Horowitz Album: Great Pianists of The Century Vol.1	Title: Arno Babadjanian Bilder for Piano Volk Artist: Yuri Egorov Album: Yuri Egorov &nc Album: Joseph James: Concerto For 3 Bouz	Title: J.S. Bach Prelude And Fugue No.10 In I Artist: Christiane Jaccottet Album: Bach: Das Wohltempierte Klavier, Te
Title: Claude Debussy Images oubliées: III. Artist: Zoltan Kocsis Album: Debussy - Suite bergamasque, Imax	Title: Camille Saint-Saens - Danse macabre Artist: Charles Dutoit - Philharmonia Orches Album: Saint Saens: Danse Macabre/Phaeth	Title: Nikolaos Skalkottas - Zalongos Dance Artist: Nikolaos Skalkottas Album: Joseph James: Concerto For 3 Bouz	Title: Manuel Ponce - Suite for guitar in A (à Artist: Segovia, Andres Album: The Legendary Segovia
Normal			
Title: J.S. Bach St. Matthew Passion, BWV 24 Artist: Monteverdi Choir &nc Album: St. Matthew Passion, BWV 244	Title: Wolfgang Amadeus Mozart Fantasy in f Artist: Alfred Brendel Album: Great Pianists Of The Century Vol.5	Title: Sergei Prokofiev Piano Concerto 3, Sz. Artist: Martha Argerich (piano), Orchestre Syr Album: Prokofiev, Bartók: Piano Concertos	Title: Frédéric Chopin Concerto for Piano &nc Artist: Yossi Shomer - Hans Zanatelli &nc Album: Chopin
Don't Like			

Figure 5. Organization Using Spatial Layout to Indicate Preference

Table 1 lists the labels for collections and lists created by participants as well as the labels for all the playlists created. Participant 4 did not create playlists due to a time constraint. These descriptors refer to user preferences, characteristics of the music, and characteristics of the activity or situation for listening to the music.

Seven participants' organizations included both positive and negative descriptors of their preference for songs. One participant did not include a positive descriptor but had a "don't care/dislike" collection. The other four participants did not express preferences in the labels attached to their organizations. Seven participants included descriptors related to musical features in their organization. Six of these included characterizations of the mood of the music ("serious", "peaceful", "calm", "aggressive")

Table 1. The Textual Descriptors Used for Collections and Playlists  
by Study Participants

Participant	Genre	Collections / Labels in Organization	Labels for Playlists
1	rock	“calm background (programming)”, “exciting (driving)”, “exciting (housework)”, “calm active (driving)”	“get there slow (relaxing, bursts of energy)”, “get there fast (keep up the energy, unwind at the end), “get moving in the morning”
2	classical	“calm”, “gloomy day”, “sunny day”, “funky”	“playlist1”, “playlist2”, “playlist3”
3		“aggressive/expositional”, “gravitas”, “ethnic/folk”, “expression”, “simplicity”	“music as art”, “music as narrative”, “dinner music”
4	rock	“love/romantic”, “lonely/forlorn”, “story”, “punk”, “philosophical”, “hmmm”, “don’t care/dislike”, “????”	N/A
5	classical	“alert but relaxed”, “driving”, “cooking and cleaning”, “entertainment” “off to sleep”, “study/curious”, “evening gathering”	“trip home”, “friends over”, “making dinner”

Table 1. Continued

Participant	Genre	Collections / Labels in Organization	Labels for Playlists
6	classical	“favorite”, “peaceful”, “delightful”, “normal”, “don’t like”	“for study”, “for party”, “for dinner time”
7		“known-like”, “unknown-dislike”, “unknown-maybe”, “unknown-like”	“list 1”, “list 2”, “list 3”
8	rock	“songs I know”, “remind me of my wife”, “cheesy”, “too slow”, “like but hard to listen to”, “fun songs” divided into “classics”, “surprised”, “hate”, “songs I don’t” divided into “generally like the artist”, “neutral”, etc.	“driving alone”, “driving with the wife”, “walking on campus”
9	classical	“favorites”, “serious”, “others”, “light, joyful”, “trash”	“playlist when boring”, “playlist when tired”, “playlist (?)”
10		“best”, “next”, “2*next”	“delight”, “calm down”, “melodic”
11	rock	“stuff I would listen to” divided into “anytime anywhere”, “every so often”, “occasionally”, “stuff I probably don’t want to hear often, if at all”	“things I can listen to over and over”, “relatively random”, “all decent, interspaced”
12	classical	“favorite”, “ok”, “like it”, “do not like”	“running”, “coding”, “reading books”

and three included terms that relate to genres (“funky”, “ethnic/folk”, “punk”). Finally, three of the participants had organizations that included labels to the contexts in which they would want to listen to music (“programming”, “gloomy day”, “off to sleep”). Overall, the personal interpretation of these spaces is consistent with the “idiosyncratic

genres” found in Cunningham and colleagues’ ethnographic study [Cunningham et al. 2004].

In the post-task questionnaires and interviews, participants detailed their strategy for organizing songs into collections and categories. 83% (10 of 12) of the participants reported grouping music based on its dynamics, especially its tempo (beat), energy, harmonic structure and tonality. Multiple participants also reported that they created collections based on how well they knew the songs (33%, 4 of 12) and how serious/important the songs sounded (25%, 3 of 12).

With regard to creating playlists, 83% of the participants reported that music dynamics were important to placing music into playlists. Some participants (25%, 3 of 12) said that they formed playlists based on the lyrics (e.g., if there are lyrics, what the lyrics say, and if they are “singable”).

Besides the creation of collections and placing text labels in the workspace to describe groupings, participants expressed their opinions about music visually. The visual attributes used most were background color (58%) and border thickness (33%). Background color was used to distinguish songs with different dynamics, express categories and preference, and to indicate familiarity with the song. Border thickness was used to indicate familiarity with the song, express preference, and distinguish songs with different dynamics (mainly tempo).

Five of the twelve participants reported using the relative position of the songs (arrangement and distance between two objects) to indicate order of playback, degree of importance, and difference in dynamics. Two participants reported using absolute

position (coordinates of objects in space) to indicate preference in a specific collection or song. Figure 6 shows examples of visual expression derived from the created workspaces. The organization on the left of the figure shows a typical example of a spatial layout for expressing importance where the favorite songs have been placed in the most visible spot of the space (top left corner) while less important music has been visually limited in small size collections at the bottom. In the example in the middle, the participant has created a playlist where the list layout is used to express possibly order of playback. Finally, on the right side, color-coding has been applied to express membership to different groups. Each of the groups is entitled with an object carrying

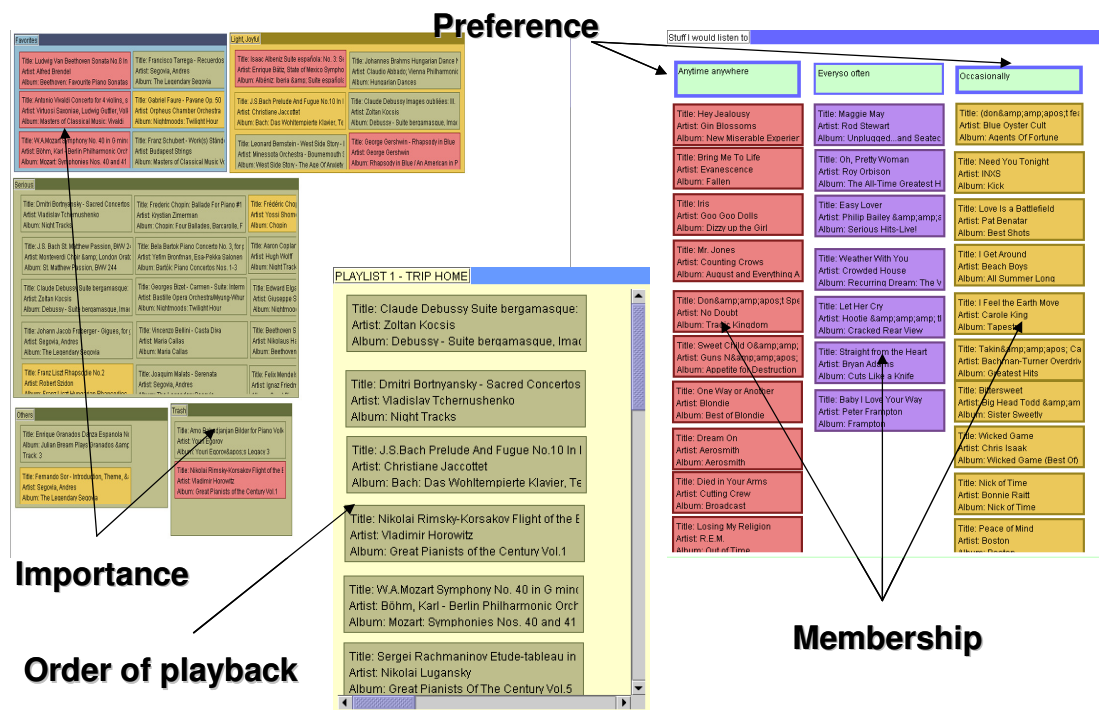


Figure 6. Visual Expression in the Workspace



Figure 7. Example Playlist

the descriptor of the music underneath. Different border thickness in the title-objects has been employed to indicate degree of preference.

Participants were asked to avoid using metadata for the organizational part of the study only, and not during the playlist creation portion of the study. The resulting playlists indicate that participants chose to put together music that they find related to (e.g., put in similar portions of the space and/or have similar visual attributes) but that are not necessarily similar in terms of explicit metadata values. Figure 7 shows a playlist of classical music that participant 6 put together for a party. Creating playlists is neither a process of collecting songs that share the maximum number of attributes nor a process



of random selection. Participants indicated creating playlists requires the selection of items that sound good and fit well together. This requirement for playlists explains why only one subject reported being happy with the metadata filtering-based playlist creation approaches found in the music software that they use.

#### *4.3.3 Comments on System Features*

While they liked using visual expression for organizing music, participants still wanted to interact with collections based on the metadata values and the explicit associations between the songs they manage. Some participants indicated the need for having interactive hierarchical/tree views of the music collection as it is stored in the file system, similar to current commercial music management software. Participants expressed an appreciation of the visibility of metadata information in the music objects in VKB as it supported a first, gross assessment of what could possibly sound good together without having to listen to the songs.

Consistent with the experiences of previous VKB users, participants' comments indicated that the VKB workspace is superior in expressive power and freedom compared to the traditional hierarchical, folder-like views of the file system. They were able to create abstract structures, express granularity in their associations, describe various types of relationships (other than similarity), and even create alternative views of the same original collection in the same workspace. Moreover, 66% of the participants said that VKB helped to organize the songs efficiently and 83% enjoyed the task.

Participants liked the preview feature of the workspace where they could listen to the first few seconds of the songs by hovering the mouse cursor over the objects.

Comments and observations show that the preview feature proved beneficial for helping users identify songs they already knew. However, playing the first 10 seconds provided by VKB was not sufficient for becoming familiar with new songs.

#### 4.4 Implications for System Design

The results of the study show that there are benefits and weaknesses to organizing personal music collections based on the context-independent metadata found in current tools and the malleable personalized interpretation found in spatial hypertext systems. This section includes a discussion of the study results and their implication for the design of future music management environments.

##### 4.4.1 *Supporting Personal Interpretation*

Knowing ahead of time what characteristics of music are going to be important to a particular user is difficult. Most music management systems support personal interpretation through the addition of new metadata fields and values. Users rarely do this because of the effort required in the human-computer interface and because users may wish to express characteristics that are difficult to describe textually (attribute names and values are generally textual.) Expressing that Blondie's *Rapture* is kind of funky but not as funky as The Sugarhill Gang's *Rapper's Delight* via metadata changes personal interpretation into a form of knowledge engineering.

Such relative assessments of musical characteristics were part of why participants positively assessed the ease of expression in spatial hypertext for personal interpretation. They found visual expression facilitated their interpretation of mood, memories, and musical dynamics. Yet, participants also indicated that the lack of views

of their collection based on traditional metadata made it more difficult to locate songs that they knew they wanted. Visual personal interpretation, at least in the time-limited task of the study, enhanced users' expression but the resulting expressions were not always efficient representations for locating specific songs.

#### 4.4.2 *Metadata Visibility, Access and Manipulability*

Systems need the predictability, consistency, and formality found in the context-independent metadata fields associated with music files. This is the strength of current commercial applications. Eight participants in the study indicated the need for having access to views of the collection based on metadata through either metadata filtering or metadata-based tree views.

The personal interpretation found in the VKB collections were based on subjective characteristics far more sensitive to change than an organization where membership is decided based on well-defined and consistent criteria. Music understanding, perception and mood are volatile factors that differ not only from person to person, but also from time to time. For example, Vivaldi's *Summer* from *The Four Seasons* may be perceived as happy in one context (e.g., a wedding) and melancholy in another (e.g., a dance party). An organization relying only on user perception can be so dynamic that locating specific pieces requires remembering the context in which the music was positioned in order to predict where it can be found. This is far more complicated than filtering explicit attributes in metadata based classifications.

The study also found that users view traditional metadata as insufficient for expressing their desires for playlists – only one of twelve participants used metadata

filters to define playlists. Participants reported that playlists involve selection of music that includes variation yet fits well together in the current context. Six participants reported that they found visual expression in VKB useful as compared to their prior experiences organizing music collections.

These results indicate that the traditional metadata (artist, composer) is valuable for navigation of a collection but not for the direct specification of desired music and that the personal interpretation found in the visual expression was valuable for selection of music but not for navigation within the collection.

#### *4.4.3 Combining User Interpretation and Context-Free Metadata*

What is missing are environments that combine the easily expressed interpretations of music found in spatial hypertext systems with the predictable and consistent explicit descriptions found in current metadata-based systems. Based on the results of the study, we currently are designing and developing a personal music management environment that integrates these two views of music collections.

Besides providing dual views of music collections based on traditional metadata and personal interpretation, this environment will attempt to bridge the gap between personal interpretation and features of music that the system can interpret. In addition to metadata, systems can assess music similarity based on signal processing of the audio content [Aucouturier and Pachet 2002; Foote 1997; Logan and Salomon 2001], collaborative filtering [van Breemen and Bartneck 2003; Crossen et al. 2002; Li et al. 2004], and lyric analysis [Logan et al. 2004]. Such techniques provide alternate, and potentially divergent, assessments of music similarity.

Spatial hypertext systems like VKB include spatial parsers that employ heuristic techniques to recognize the interpretive structures created by users [Francisco-Revillia and Shipman 2004]. The recognized visual structures can indicate what music characteristics the user finds relevant for their organization. These characteristics can then be the basis for computing a personalized clustering of music collections or for personalized weighting in relevance feedback algorithms [Hoashi et al. 2002].

#### 4.4.4 *Easy Access to Music*

Creating and managing collections based on how music sounds requires sufficient knowledge of the music content. Organizing a small set of familiar songs can be an easy task of simply remembering and associating the basic melodies. However, classifying a large quantity of music, such as people collect over years, can be a challenging and time-consuming process requiring extended periods of listening to and comparing songs. The study showed that having direct access to the music content without the contextual overhead of launching additional applications simplifies the process. Access to short snippets of music supports people's remembering what a song sounds like while access to the music as a whole is beneficial for assessments of unfamiliar music. To improve the efficiency of access, the environment will generate and play music summaries [Logan and Chu 2000; Cooper and Foote 2002], which may provide more time-efficient overviews of the musical content.

#### 4.5 Preliminary Study Summary

Participants in the preliminary study provided a wide variety of personal interpretation when using a general purpose spatial hypertext to organize a music

collection. While they were positive about this ability, participants desired more access to the collection through traditional metadata as well. Finally, the audio preview provided in VKB was found valuable for identifying known music but was insufficient for getting a sense of unknown music.

## 5. MUSICWIZ DESIGN

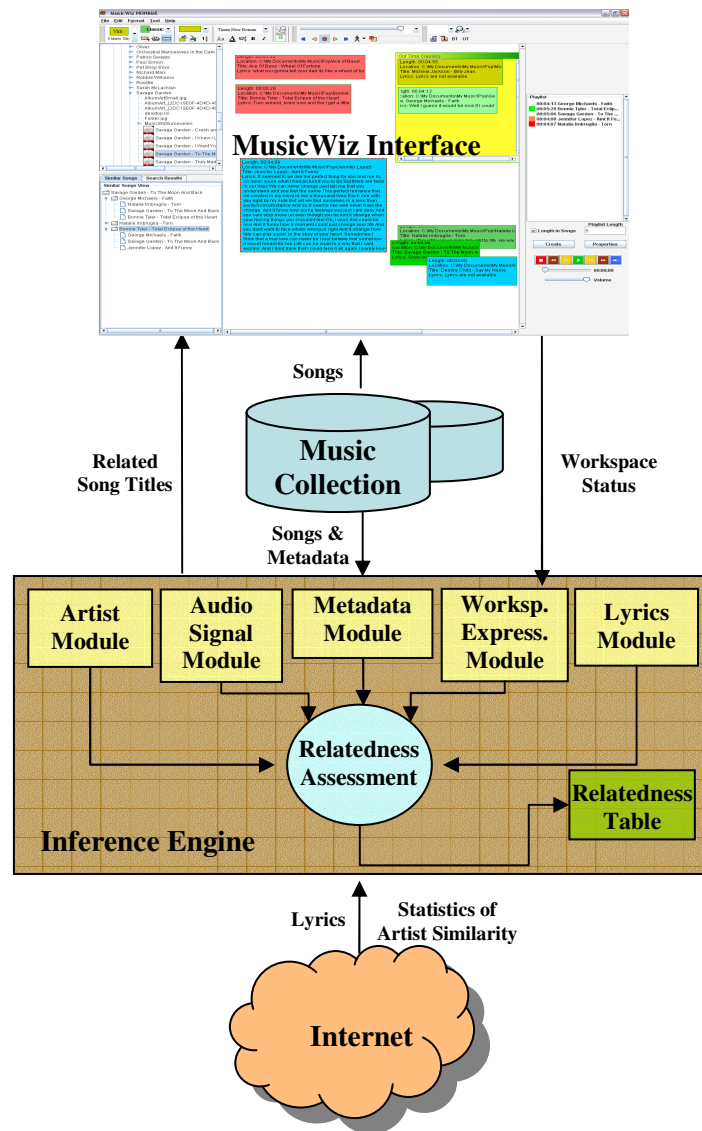


Figure 8. MusicWiz's Architecture

The feedback received from this formative study not only indicated the potential for non-verbal expression in music management, but also provided important information of how and under what conditions people would use such an environment for organizing and enjoying their collections. This section describes an approach for managing and playing music using human expression, metadata and inferred similarity based on feature extraction. Figure 8 shows MusicWiz’s architecture. It currently

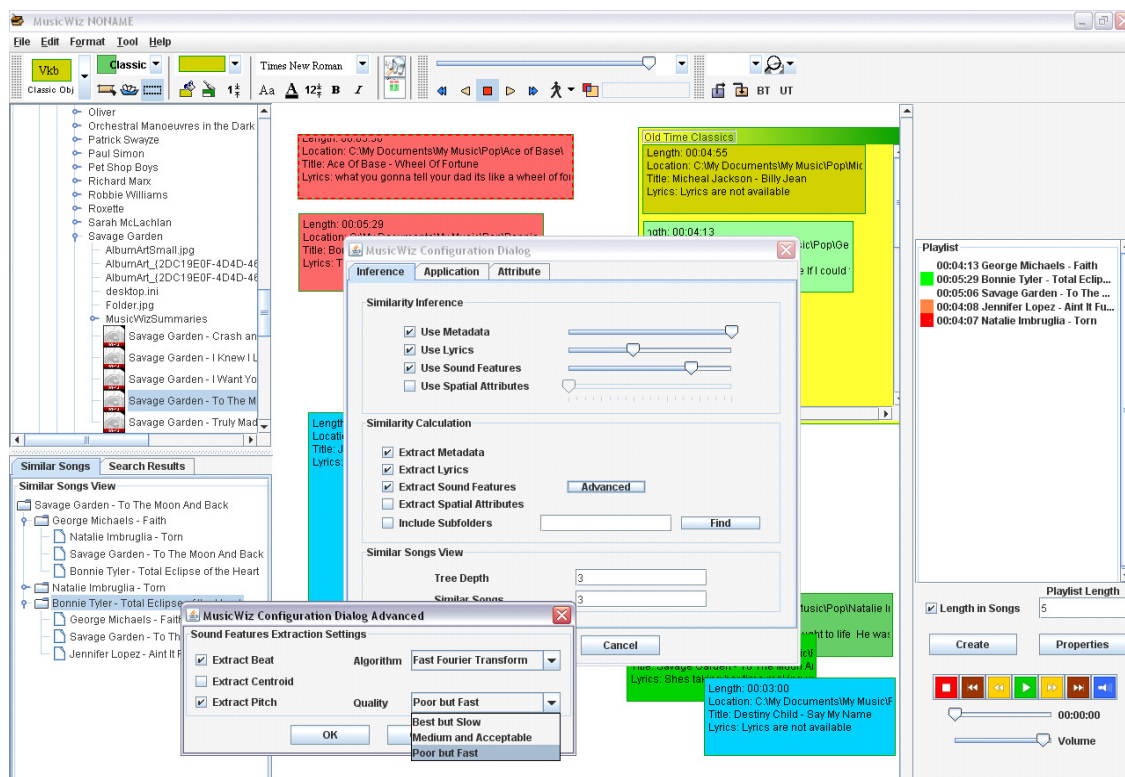


Figure 9. MusicWiz’s Interface Combines a Tree View, a Workspace, and an Area for Search Results and Related Music



consists of two major components: an interface for interacting with the music collection that supports personal expression and an inference engine for assessing music relatedness based on a combination of explicit and implicit information about music and the user's personal interpretation in the interface.

## 5.1 Interface and Functionality

MusicWiz's interface employs an information workspace similar to that of VKB alongside traditional metadata-based the-view of the collection, a region for MusicWiz to present search results and suggestions of related music and a pane for playlist creation and playback (see Figure 9).

### 5.1.1 Workspace Components and Display

Songs in the MusicWiz workspace are represented as two-dimensional components that can be changed visually and spatially to implicitly express a wide variety of relationships. Users can modify their visual attributes like the background and border color, the border thickness, the font of the text, and the width and height of the

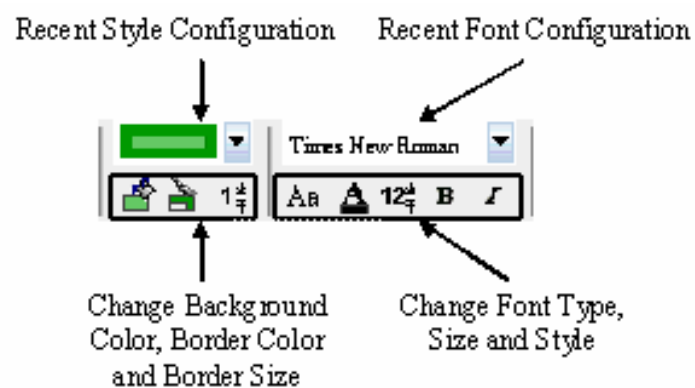



Figure 10. MusicWiz's Visual Attributes Controls

component (see Figure 10). They can also organize the components to form structures like lists, piles and composites. Studies with spatial hypertext over the years have shown that attributes and associations including importance, hierarchy, membership, degree and type of similarity are easily expressed with the right selection/combination of visual attributes [Marshall and Shipman 1995]. Accordingly, the absolute and relative position of the components in the workspace can be very informative about the degree of uniqueness or importance of the components, their membership and order, even their similarity [Shipman and Marshall 1999].

One-thing participants appreciated when organizing songs in VKB was the fact that they had direct access to the metadata values. It was easy for them to make a first gross clustering without having to access the actual music content. In MusicWiz, every object holds a rich amount of information including its ID3 tag values of the song, the location of the respective audio file in the local drive and its full lyrics.

Objects can be stored together in a collection. Following a concept similar to that of the folders in windows, collections can contain objects and other collections. A collection can be created by selecting the  button in the interface toolbar and clicking anywhere on the canvas. Changing the size, color, and border of a collection is done in exactly the same manner as changing those features in an object. The collection title is editable and can be modified by clicking on the title bar of the container. A collection can be also moved in the workspace and assigned directly with its content to other collections.


A third kind of component that can be used in the workspace is plain objects. Plain objects can be created by selecting the  button in the interface toolbar. They are general purpose information entities that can be visually and spatially edited and handled exactly as the song objects. However, they do not hold any predefined role and hence do not contain initial information. That makes them ideal for applications like annotation or titling of other workspace components as their text, in contrast to the song objects, is editable.

Figure 11 shows an example of the three types of components supported by MusicWiz. In the left side collection, a plain object in light green is used to title a two-column list of song objects that the user has grouped together as playlist material. On the right side collection, a plain object in purple is used to annotate a heap of song objects that the user has not accessed recently.

### 5.1.2 *Music Access and Retrieval*

One of the complaints participants of the preliminary study had using VKB was the unavailability of metadata-based and location-based hierarchical views of the music collection. Such conventional classifications are more consistent and resistant to change over time, which means that they can be used as a safe starting and reference point for building less conventional organizations. Additionally, such views provide efficient access with searching techniques that people are already familiar with. MusicWiz provides a tree view of the music in the collection alongside the spatial hypertext workspace (see upper part of Figures 12a and 12b).



Figure 11. Examples of Song and Plain Objects inside Collections in Spatial Formation

Beneath the file system tree view, another tree view displays songs that are similar to the currently selected songs in the system tree view (see lower part of Figure 12a). This area provides users with easy access to songs that are related to the selection based on the Inference Engines analysis. Users, having direct access to music that is related to their prior choices, can select music for playlists without having to perform multiple searches. When the user right-clicks on a song in the system tree-view and selects the option *find similar songs*, MusicWiz generates a tree with the titles of a fixed number of related songs ordered from the most to the least similar. The user then can drag and drop files into the workspace and assign them to collections or into playlists.

Each branch of the tree can expand to show another level of similarity. The songs at each level are directly related to their parent node and indirectly to the root of the tree and hence to the initial selection. Users can specify the maximum number of children per branch and the depth of the tree as well as the combination of similarity metrics used by the inference engine in a configuration dialog (top and bottom part of Figure 13).

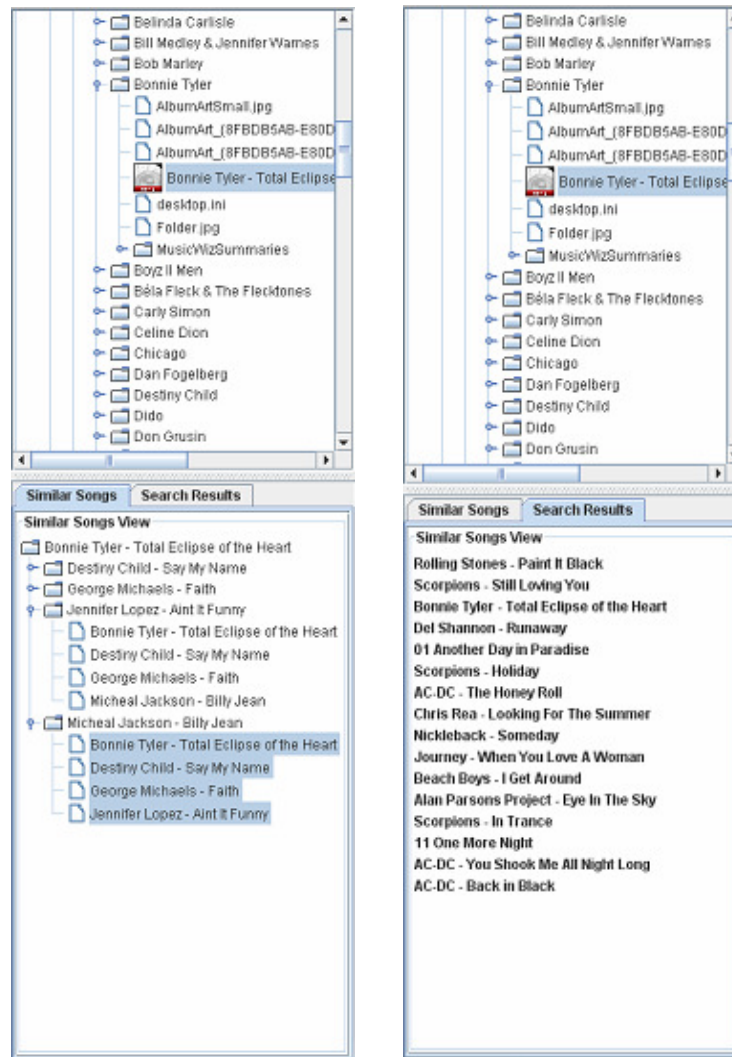


Figure 12. MusicWiz's Tree View, Similar View (a) and Search View (b)

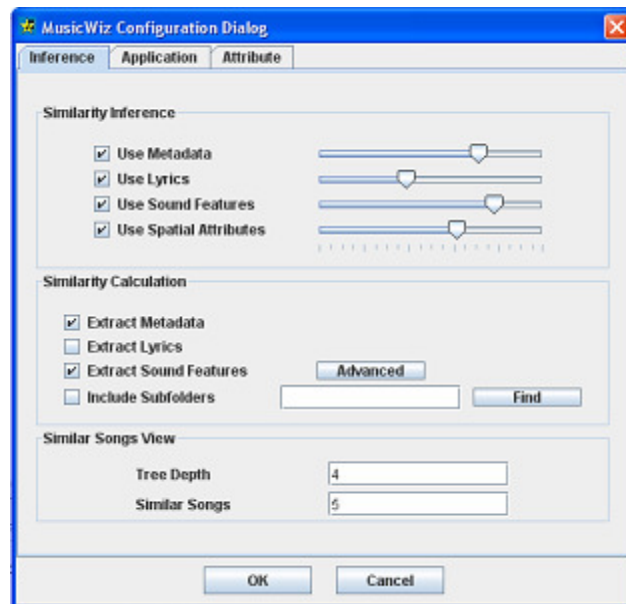



Figure 13. MusicWiz's Configuration Dialog

In addition to retrieving related music by navigating from song to song in the related songs tree view, the system also provides advanced search capabilities. Users can directly access and filter music based on a wide range of attributes including metadata values, lyrics (occurrence of a specific phrase or set of phrases), and sound and melody features. The search attributes can be easily accessed and configured in the *Song Search Menu* (Figure 14) that is triggered by pressing the  button in the interface main toolbar. The predefined values in the *Signal Attributes* fields beat and brightness reflect the minimum and maximum values of those attributes based on the songs currently in the collection. The system displays the list of the returned results as a separate tab in the same panel hosting the related songs tree view (see Figure 12b). Songs then can be

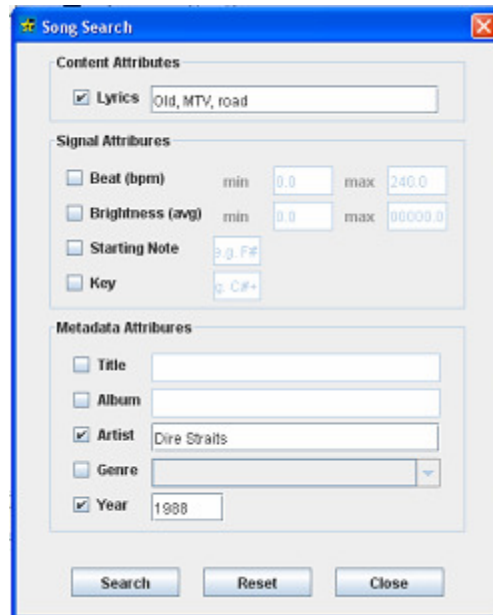


Figure 14. MusicWiz's Search Menu

dragged and dropped from the list into the workspace and the playlist pane to update collections and playlists respectively.

### 5.1.3 Music Playback and Playlist Creation

In the preliminary study, participants viewed that VKB had limited applicability as everyday software for music playback because of the need for an external application to listen to the songs they were organizing. The time and effort of switching between



Figure 15. MusicWiz's Playback Controls



Figure 16. MusicWiz's Playlist Pane

applications proved to be significant for rapidly classifying a large volume of music. MusicWiz provides full playback functionality directly in the application with controls in the lower right corner of the interface including a slider for easy within song navigation (Figure 15). Playing a song is just a matter of either dragging the file from any of the left side views (*tree view*, *similar songs view* and *search songs view*) and dropping it to the playlist pane (Figure 16) or adding it to the latter through the popup menu options provided when right-clicking on the song objects.

The creation and population of a playlist in MusicWiz can be done either manually or automatically. Creating a playlist manually is a straightforward process that basically follows similar steps to those in music playback. MusicWiz allows songs to be dragged and dropped from any of the left side views of the interface to the *playlist pane* (see Figure 13). Songs in the workspace can be also added to the playlist by right-clicking on their object and selecting the *Append Playlist* choice in the popup menu. The



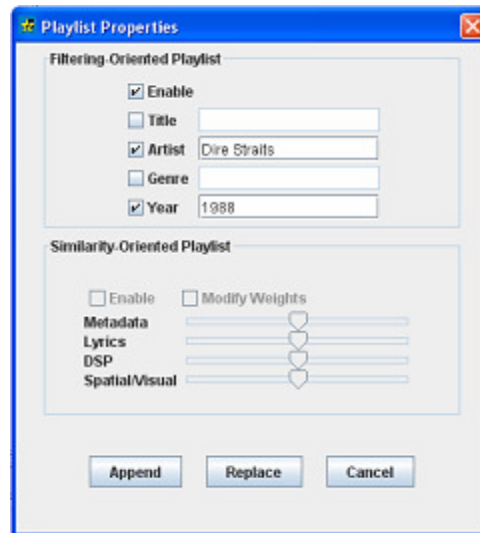


Figure 17. MusicWiz's Playlist Properties Menu

choice *New Playlist* will remove any existing playlist content before adding the new song(s).

To create playlists automatically, MusicWiz utilizes the similarity assessments stored in the inference engine. The system provides two basic modes for system-assisted playlist creation: filter oriented and similarity oriented. Both can be configured through the *Playlist Properties* menu (Figure 17) that appears when pressing the *Populate* button right under the *MusicWiz Playlist Pane*.

#### 5.1.3.1 Creating Playlists Automatically – Filter Oriented Playlists

In this mode MusicWiz selects music by "filtering" the collection based on its ID3 tags. Currently the system supports retrieval of songs according to the title, the artist name, the genre and the year of music. The *Append* button in the *MusicWiz Playlist Properties Menu* will populate the existing playlist by appending it with the new songs

while the *Replace* button will remove any existing playlist content before adding the new songs. Users can specify the total length of the new playlist generated by the system in the *Playlist Length* text box (as either number of songs or total duration in minutes), right under the *MusicWiz Playlist Pane*. They can also customize the resulted selections by adding, removing or reordering them accordingly.

#### 5.1.3.2 *Creating Playlists Automatically – Similarity Oriented Playlists*

In this mode, MusicWiz selects music that the Inference Engine considers similar to the songs of the current playlist. Specifying how and which of the existing songs will be used as "examples" for the retrieval of related music can be done through the popup menu generated when right-clicking on any of the songs in the playlist. An "example" can function as the start (*Path Starting Point* - green flag), as an intermediate / visited node (*Path Reference Point* - orange flag) or as the end of the new playlist (*Path Ending Point* - red flag). The system can retrieve related songs only if there is a green flag song. In that case, all the generated songs are similar to the green flag one ordered from the most to the least similar. A green flag song combined with a red flag one steers MusicWiz towards selections that provide a smooth transition from the green flag song (start) to the red flag song (end). Finally, the addition of any intermediate / visited nodes (orange flag songs) adds a bit more user intervention to the resulting choices by forcing the system to make selections that “bridge” those nodes as the music progresses. The type of the relation between the new songs and the "examples" can be qualitatively and quantitatively determined in the *Playlist Properties Menu*. The *Similarity Oriented Playlist* section provides sliders that can calibrate the contribution of each of the

attributes for which the similarity has been calculated to the new playlist selection. The functionality of the *Append* and *Replace* buttons is the same as in the case of the Filter Oriented Playlists appending and replacing the existing playlist content respectively according to the length in the *Playlist Length* text box. The *Recover* button in the same location undoes any dissatisfying appends or replaces causing the playlist to return to its status before the latest action. The final selections then can be further customized with reordering existing songs, adding new songs or removing existing ones.

#### 5.1.4 *Music Preview*

Although the short previews of songs proved to be useful in helping participants recall songs they already knew, they were not sufficient for participants to become familiar enough with new music to classify it in their organizations. The short duration of the snippets (10 seconds) was one reason for this. Most online music stores provide previews of 15 to 30 seconds. This is barely enough time to provide a sense of the different melodies of a song. Another reason that the snippets were not sufficient for becoming familiar with new music had to do with the choice of the content and more specifically with using the introduction for representing the entire song.

There is no doubt that a song's introduction can uniquely identify it. Who can forget the four-note opening motif in Beethoven's Symphony No.5? However, there are other parts like the refrain that can frequently provide a better overall and more recognizable representation of the music (especially in modern, western music genres where the refrain is also the most repetitive segment of the song). MusicWiz generates music summaries using a set of signal processing algorithms for automatically extracting

music phrases considered important due to their repetition and uniqueness. Each summary has a total length of 22 seconds and consists of the most salient phrase of the song (usually the refrain) supported by two additional, highly repeated parts. A complete presentation of alternate algorithms for summary generation and a study of their effectiveness is found in section 6.

## 5.2 Inference Engine

MusicWiz's inference engine supports access to the music collection through relatedness. Music can be related in many ways. It can have similar melody or sound features, be by the same artist, and have lyrics that share common themes, or convey a similar mood or feeling.

The MusicWiz inference engine consists of several modules that are responsible for extracting, representing, and comparing information about the songs to assess their relatedness. Currently, there are modules for processing and comparing artists, metadata, audio signals, lyrics, and workspace expression. Each of these modules produces an assessment of relatedness (a normalized value ranging from 0 – songs very dissimilar, to 1 – songs almost identical) that is combined by the Inference Engine for evaluating the overall relatedness of the songs in the collection. The following subsections describe the assessment of similarity by each of these modules.

### 5.2.1 *Metadata Module*

The main task of the *metadata module* is the evaluation of similarity in the metadata values. Field comparison is applied to every possible pair of songs in the collection. The module reads the ID3 tags and the location of the music files in the local

drive and then performs string comparison using a distance metric that combines the Soundex [Knuth 1973] and the Monge-Elkan [Monge and Elkan 1996] algorithms. The Soundex phonetic algorithm is valuable for identifying similarity between transliterated or misspelled names. It uses the six phonetic classifications of human speech sounds to convert the input into a string that identifies the set of words that are phonetically alike (similar pronunciation). The Monge-Elkan algorithm identifies similarity among expressions where the words are listed in a different order. It is a dynamic programming algorithm that in general terms calculates the distance of two strings based on the cost of transformations required to convert the first expression into the second expression. The string comparison in the *metadata module* is applied to the title, artist, genre, year, and album-name of the songs as well as the file-system path where they are stored. In the case of the first four tags, the system evaluates similarity by taking the average of their Soundex and Monge Elkan distance. The year and file-system path values are compared exclusively based on the Monge Elkan metric.

Once individual metadata fields have been compared, these assessments are combined for an overall rating of the relatedness of two songs based on their metadata. The module currently combines the field comparison results using equal weights to calculate the overall metadata similarity.

$$\begin{aligned} \text{Overall Metadata Similarity } (S_1, S_2) = & \sum WA_n * (SN(S_1(A_n), S_2(A_n)) + \\ & + ME(S_1(A_n), S_2(A_n))) / 2 + \sum WB_k * ME(S_1(B_k), S_2(B_k)), \end{aligned}$$

where  $S_1, S_2$  are the songs under comparison,  $WA_n$  the weight of the  $A_n$  tag value ( $A = \{\text{title, artist, genre, album-name}\}$ ,  $n = 1 \dots 4$ ),  $WB_k$  the weight of the  $B_k$  tag value ( $B = \{\text{year, path}\}$ ,  $k = 1 \dots 2$ ), SN the Soundex distance and ME the Monge-Elkan distance.

### 5.2.2 *Audio Signal Mode*

The *audio signal module* relies on digital signal processing (DSP) to compare characteristics of the two audio signals. It consists of several algorithms that process the sound waveforms of the songs and extract information about their harmonic structure and acoustic attributes. Currently, the module includes algorithms for extracting the beat (tempo), the brightness (centroid), the pitch (fundamental frequency) and the potential key (music scale) of the song. The greater the distance in the beat, brightness and pitch levels, the less likely they are perceived as being of similar style or mood.

#### 5.2.2.1 *Beat Extraction*

MusicWiz provides two options for extracting the beat of a song. Their basic difference is in the way the frequency components of the signal are calculated.

The simpler of the two uses the Fast Fourier Transform or FFT approach. In the preparation phase the system discards the first and last 20 seconds of the song as they usually lack the information (e.g. rhythmical patterns) that can be used for the beat estimation. The signal then is downsampled from 44.1kHz to 630Hz (the focus is on the frequencies of the lower band) and an autocorrelation function is applied to it for the detection of any repeating components (experimentation with low pass filters and successive applications of the autocorrelation for further simplification and “cleaning” of the signal didn’t show any significant improvement in the algorithm’s accuracy). The

output is smoothed out repeatedly by applying a cubic spline to the minima and maxima until the standard deviation in the output between successive siftings becomes less than a threshold (i.e. 0.3). Next, a Fast Fourier Transform (FFT) with a Hann window is used to determine the frequency components of the processed signal and identify peaks (maximum amplitude) in the frequency range where the beat will likely occur (currently limited between 0.3 and 2.5Hz or 18 and 150bpm).

Figure 18 shows an example of the different stages of the FFT-based approach as it is applied in C. Isaac's *Wicked Game*. Starting from left to right and top to bottom, we can see a snapshot of the original signal (raw waveform), a couple thousand points from the output of the autocorrelation function (notice the periodic peaks, especially those in lags 680 and 1360), the output of the autocorrelation function smoothed-out with the application of the cubic spline on the maxima, and finally the results of the FFT in terms of magnitude over the frequency range of interest. The maximum peak at 0.94Hz ( $\approx$  56 bits/min) indicates the beat, something that can be also confirmed from the second strongest peak at 1.88Hz.

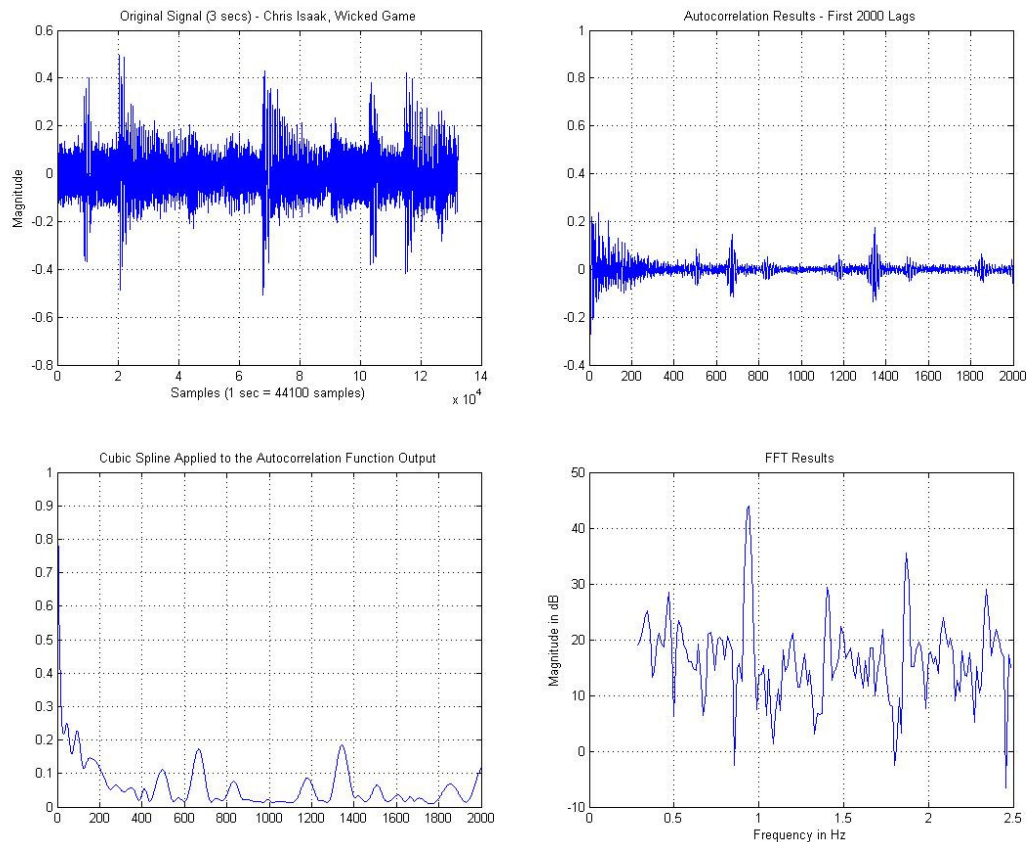


Figure 18. FFT-based Approach for Beat Extraction of C. Isaak's *Wicked Game*

Performance-wise, the FFT-based approach behaves acceptably well requiring about 10 seconds to process and extract the beat of a 60MB wav file on a 2.4GHz Intel Core 2 processor with 2GB of RAM.

The second approach for the beat detection uses the discrete wavelet transform or DWT analysis for the detection of the signal frequency components. Compared to the FFT technique, the DWT is in theory computationally less demanding ( $O(N)$  time as compared to  $O(N \log N)$ ) and offers superior performance when the signal contains sharp spikes or discontinuities. In the preparation phase, the system discards 20 seconds from



the introduction and loads the next 100 seconds of the song. Without downsampling, the signal is then divided into successive, non-overlapping blocks of 1-second length each (usually 44,100 samples). For each of the blocks the system performs a first level – one dimensional wavelet analysis using the Daubechies 2 (db2) mother function. In wavelet transform, the mother or prototype function (wavelet) is a fast-decaying oscillating waveform that is scaled and shifted to match the signal for the discovery of its frequency components in time. The detailed coefficients of all the blocks are merged into a single

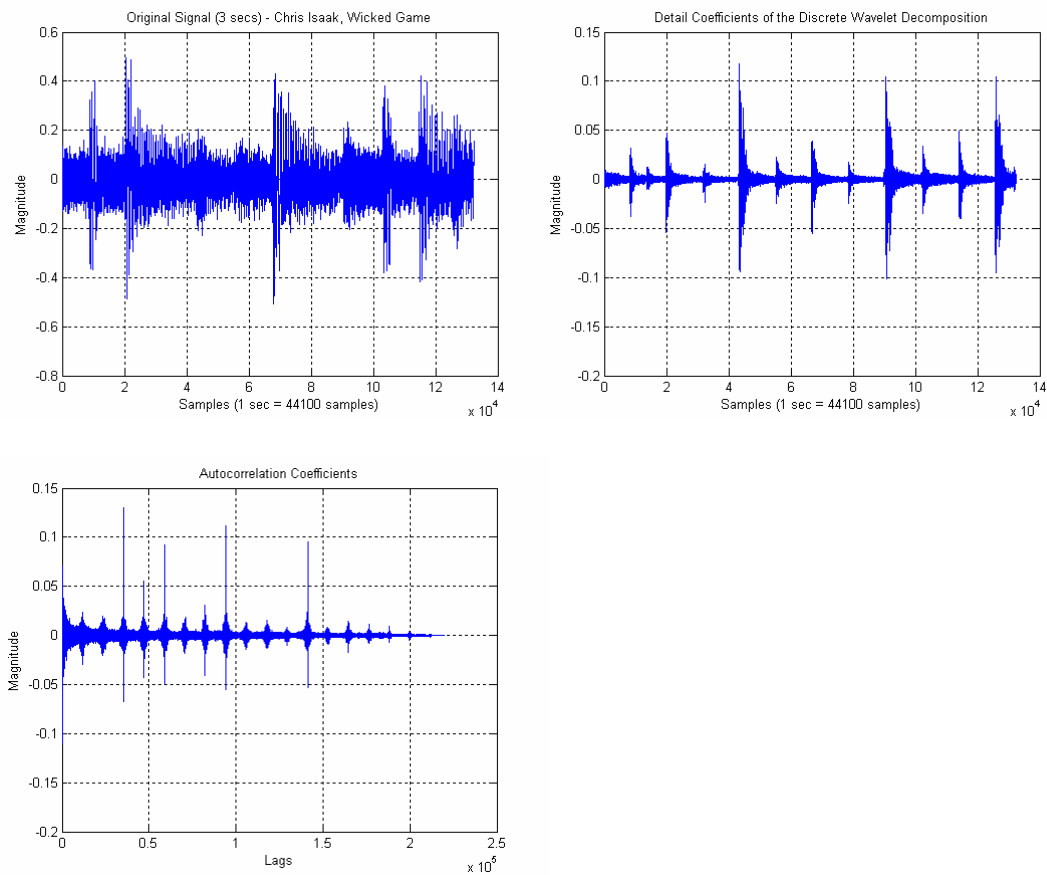


Figure 19. DWT-based Approach for Beat Extraction of C. Isaak's *Wicked Game*

stream that is used as input to an autocorrelation function for the search of any periodic components. Next, the output of the autocorrelation function is used for the detection of any equally distant peaks (maxima that occur within a specific distance threshold in time) in a range of frequencies between 0.5 and 2.5Hz. The frequency of points of the most consistent sequence is then the beat of the signal.

Figure 19 shows an example of the different stages of the DWT-based approach as it is applied in C. Isaac's *Wicked Game*. Starting again from left to right on the top row, we can see a snapshot of the original signal (about 3 seconds) and the first level detail coefficients of the discrete wavelet decomposition. Notice that the distance between successive peaks of similar amplitude is approximately 42,000 to 43,000 samples which is something less than 1 second. That is consistent to the beat of the song which is about 0.94Hz. The last graph contains the output of the autocorrelation function (notice the periodicity in the spikes) that exposes the periodic components in the stream of the detailed coefficients. Although it doesn't seem necessary in this case, the use of the autocorrelation function is particularly helpful when the wavelet coefficients cannot be interpreted in such a straightforward manner. Figure 20 shows an example where identifying the periodic components is perhaps easier analyzing the autocorrelation coefficients on the bottom left graph than the detail coefficients on the top right one.

Performance-wise, the DWT-based approach requires about 1 minute and 10 seconds to process and extract the beat of a 60MB wav file on a 2.4GHz Intel Core 2 processor with 2GB of RAM. This is much worse than the FFT-based approach and due to the fact that the signal is processed without downsampling for precision reasons. In

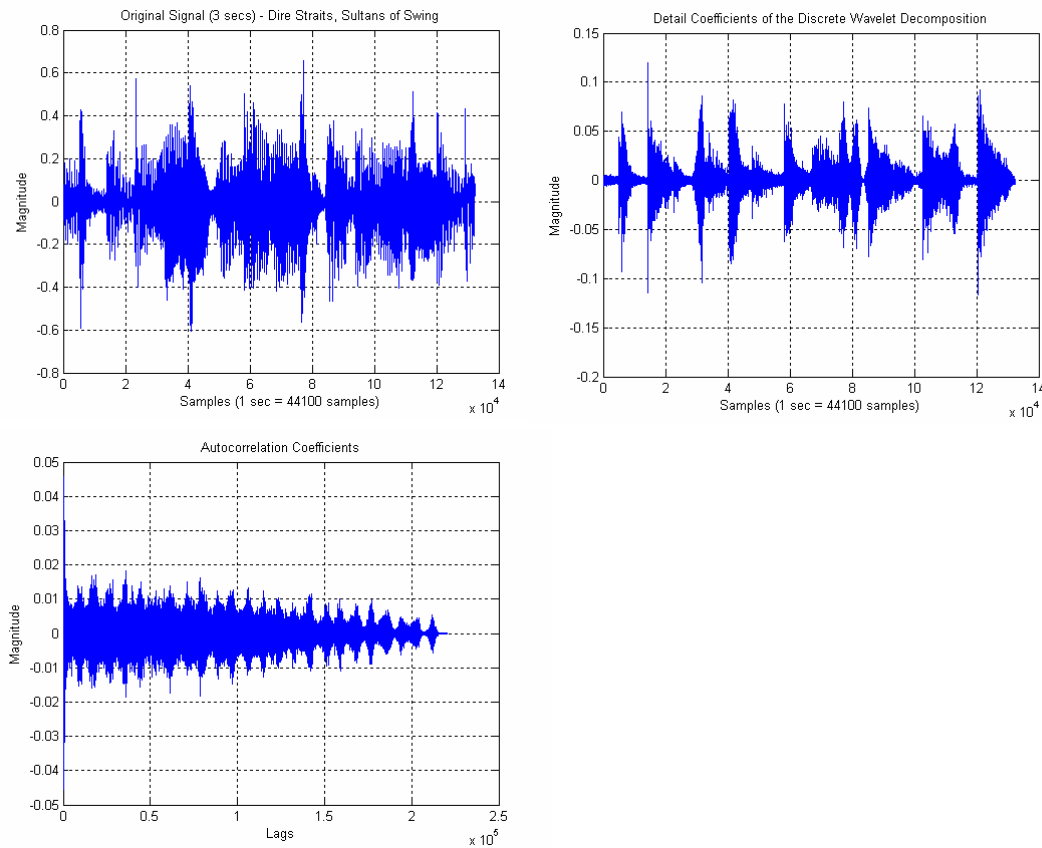


Figure 20. DWT-based Approach for Beat Extraction of Dire Straits, *Sultans of Swing*

fact, the DWT-based approach returns more accurate results when compared to the FFT-based algorithm but only in genres where the beat is not as distinct as in rock or pop music (e.g. jazz or classical). The user can select the approach to beat extraction used. The module uses the FFT-based approach by default.

#### 5.2.2.2 Beat Comparison

Regardless of the approach is used to extract the beat, MusicWiz calculates the beat similarity of two songs  $S_1$  and  $S_2$  by taking the ratio of their beats:

$$\text{Beat Similarity } (S_1, S_2) = \text{Min } (\text{Beat}(S_1), \text{Beat}(S_2)) / \text{Max } (\text{Beat}(S_1), \text{Beat}(S_2))$$

Hence, if the tempo of  $S_1$  is 60bpm and the tempo of  $S_2$  80bpm respectively, then their beat similarity is 0.75.

### 5.2.2.3 *Brightness Extraction*

The brightness of a song is strongly related to the centroid of the sound. Centroid is a popular psycho-acoustical feature that quantifies the mean frequency range of the signal in relation to the amplitude. In simple terms, it measures the position in Hz of the center of mass of the signal's frequency spectrum. The higher the centroid is the brighter the signal sounds to the human ears.

To calculate the brightness, the system removes first 20 seconds from the start and the end of the signal to prevent the introduction of noise from any silent or non representative parts of the song. The signal is then segmented into fixed-size chunks of 1-second length each with no overlap. A FFT is applied to determine the frequency spectrum of each chunk which in turn is used to estimate the brightness in every single second of the song [Annesi et al. 2007]. The brightness calculation formula takes the weighted mean of the frequencies present in the signal (basically the center frequency of each of the FFT bins), with their magnitudes as the weights. The algorithm returns the maximum brightness of the song as well as the sequence of brightness value every six seconds with 50% overlap.

Figure 21 shows two examples of brightness fluctuation in Carlos Santana's songs *Oye Como Va* and *Evil Ways*. The peak just after the second minute on the left graph reflects the beginning of the organ solo that starts from a high G (6<sup>th</sup> octave) with

a very rich and bright sound and continues even higher and finally finishes with the introductory repeating pattern in A4 (3<sup>rd</sup> octave). In the second graph the organ solo begins right after the 80<sup>th</sup> second. However, it is in a mid and low frequency range where the instrument by its nature sounds dull and hypotonic. That explains why the brightness level is low all the way up to 105<sup>th</sup> second. After that point the solo moves to higher frequencies with chords and vocals that make the sound bright.

The brightness extraction algorithm requires less than 17 seconds to process and extract the brightness of a 43MB wav file on a 2.4GHz Intel Core 2 processor with 2GB of RAM. This is similar to the time for the FFT beat extraction algorithm but significantly faster than the wavelet approach to beat extraction.

#### 5.2.2.4 *Brightness Comparison*

To determine the brightness similarity, MusicWiz employs two metrics: the maximum brightness and the average brightness. First, the system evaluates the

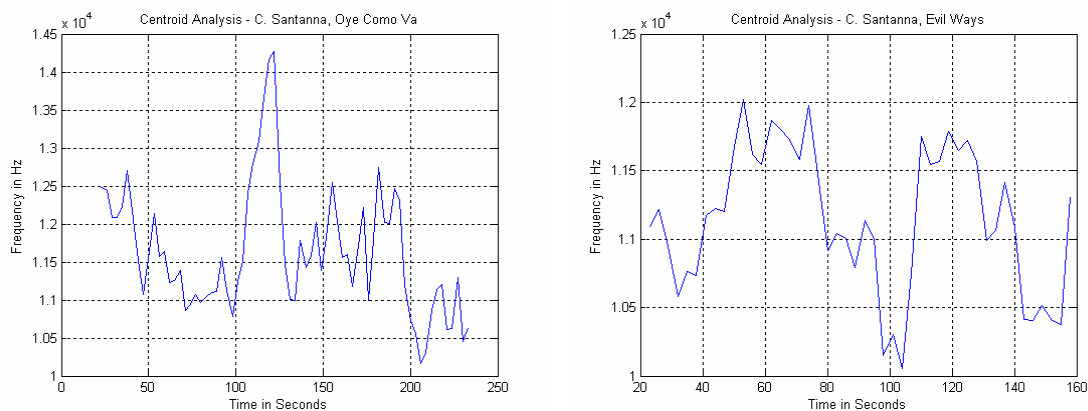


Figure 21. Brightness Levels over Time for Two C. Santana's Songs

maximum and average brightness similarity of the songs by calculating the following ratios:

$$\text{Maximum Brightness Similarity } (S_1, S_2) = \text{Min } (\text{MaxBrightness}(S_1), \text{MaxBrightness}(S_2)) / \\ / \text{Max } (\text{MaxBrightness } (S_1), \text{MaxBrightness } (S_2))$$

$$\text{Average Brightness Similarity } (S_1, S_2) = \text{Min } (\text{AvgBrightness}(S_1), \text{AvgBrightness } (S_2)) / \\ / \text{Max } (\text{AvgBrightness } (S_1), \text{AvgBrightness } (S_2))$$

Then, it calculates the brightness similarity by taking the average of the maximum and average brightness similarity:

$$\text{Brightness Similarity } (S_1, S_2) = (\text{Max Brightness Similarity } (S_1, S_2) + \\ + \text{Average Brightness Similarity } (S_1, S_2)) / 2$$

#### 5.2.2.5 Pitch Extraction

The pitch is a subjective psychophysical attribute of the sound that has to do with how humans perceive musical tones. It is strongly related to the harmonics of a sound and especially the lowest harmonic known as fundamental frequency or  $F_0$ .

For the calculation of pitch, the MusicWiz utilizes the autocorrelation-based algorithm YIN [Cheveigne and Kawahara 2002] that combines good performance with low error rates and no upper limit on the frequency search range (good for songs with high frequencies). In the preprocessing phase the system loads the wave file and segments it into fixed-size blocks of 1.5 seconds length each. Then, depending on the requested level of speed (users can select among three options that gradually decrease the speed of the process to benefit the precision of the results – see Figure 22), the system discards a predefined number of blocks to accelerate the processing of the signal



Figure 22. Three Levels of Accuracy / Speed for Pitch Extraction

in the following steps. In the medium quality option, the algorithm keeps only the first 75 seconds and the last 15 seconds of the music (based on the assumption that the signal's fundamental frequencies are usually closer to those of the scale (key) the song is composed in found in the introduction and at the ending part) while in the worst quality option only the first 30 seconds. The best quality option utilizes the full signal and does not discard any blocks. In the processing phase, the system uses the YIN algorithm to estimate the fundamental frequency  $F_0$  of each block and hence the pitch of the sound every 1.5 second of the song. The set of all the pitches is then compared to the frequencies comprising the traditional western, major and minor, music scales. The best match (minimum number of errors of type the occurred pitch not being a pitch of the scale) is recognized as the potential key of the song. Other useful features this process returns include the starting pitch of the song as well as the five most frequent fundamental frequencies.

Figure 23 shows the pitch analysis (fluctuation of musical tone over time) in the best quality option for four pop and rock songs. A good example of the descriptive power of the algorithm can be seen in the top left figure (Sting's *Shape Of My Heart*) where the high-pitch harmonica arpeggio that starts from a F5# in 3:41 and completes in

a C6# at 3:45 causes that distinct spike around the 225 second of the graph. In C. Santana's *Evil Ways* at the bottom right figure, the set of successive maxima starting in the 28<sup>th</sup> second of the graph and repeat almost identically in the 66<sup>th</sup> second accurately capture the climax of the repetitive theme with the vocals and the organ in the high frequency range.

Performance-wise, the medium option seems to provide the best compromise between speed and accuracy. The best quality option does not seem to improve the precision that much and is significantly slower. On the other hand, the worst quality

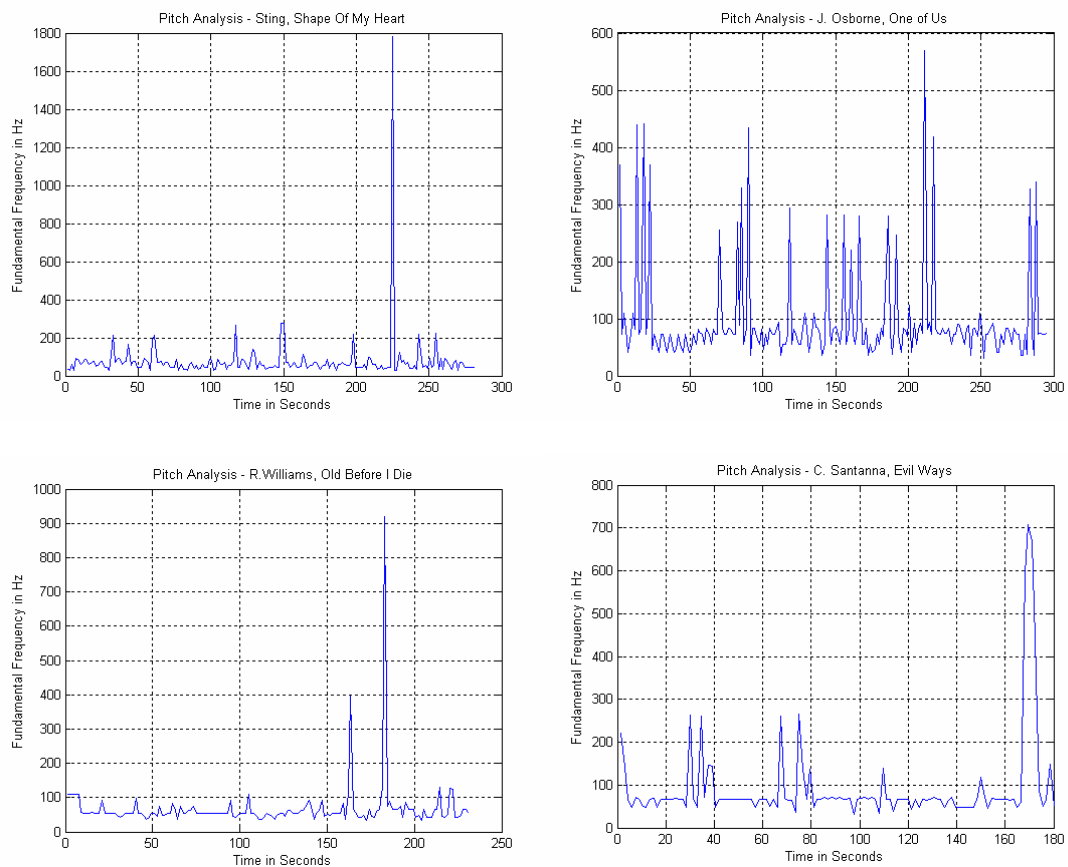


Figure 23. Pitch Analysis of Four Songs in the Best Quality Option



choice is far less demanding but cannot compete in precision with the other two. However, it can be a good choice when the speed is an issue. Tests showed that processing a 45MB wave file on a 2.4GHz Intel Core 2 processor with 2GB of RAM takes about 8 minutes and 30seconds in the best quality mode, 2min and 50secs in the medium quality mode, and finally about a minute in the worst quality mode.

#### 5.2.2.6 Pitch Comparison

To calculate the pitch similarity, MusicWiz first determines the similarity in the five most frequent fundamental frequencies of the songs:

$$\text{Fundamental Frequency Similarity } (S_1, S_2) = \frac{\# \text{ of overlapping } F_{os} (S_1, S_2)}{\# \text{ of } F_{os} \text{ per song}}$$

Then, it determines the similarity in the potential key that the songs are written:

$$\text{Music Key Similarity } (S_1, S_2) = \frac{\# \text{ of overlapping potential keys } (S_1, S_2)}{\# \text{ of potential keys per song}}$$

Finally, it checks if the songs start from the same music note:

$$\text{Starting Note Similarity } (S_1, S_2) = \begin{cases} 1, & \text{if it is the same} \\ 0, & \text{if it is not} \end{cases}$$

The pitch similarity then of the two songs is defined as the following averaged sum:

$$\text{Pitch Similarity } (S_1, S_2) = (\text{Fundamental Frequency Similarity } (S_1, S_2) + \text{Music Key Similarity } (S_1, S_2) + \text{Starting Note Similarity } (S_1, S_2)) / 3$$

### 5.2.2.7 Overall Audio Signal Similarity

After the individual values for the beat, brightness and pitch similarity have been calculated, MusicWiz determines the overall audio signal similarity of the songs by the taking the following weighted sum:

$$\begin{aligned} \text{Overall Audio Signal Similarity } (S_1, S_2) = & W_1 * \text{Beat Similarity } (S_1, S_2) + \\ & + W_2 * \text{Brightness Similarity } (S_1, S_2) + W_3 * \text{Pitch Similarity } (S_1, S_2), \end{aligned}$$

where  $W_1$ ,  $W_2$ , and  $W_3$  the contribution of each of the sound attributes.

### 5.2.3 Lyrics Module

The *lyrics module* uses textual analysis of the lyrics to identify similar songs. Lyrics are scraped from a pool of popular websites and stored in the local database for either display in the objects of the workspace or processing and comparison. To assess the lyrical similarity of two songs, MusicWiz generates their term vectors and calculates their cosine similarity [Salton et al. 1975]. The larger the number of the common representative words in the lyrics, the greater the possibility the songs to be motivated by or to describe related themes.

### 5.2.4 Workspace Module

The *workspace expression module* includes a spatial parser that identifies relations between the components of the information workspace based on their visual attributes and spatial layout. Studies with spatial hypertext over the years have shown that, while expression in this kind of workspaces is unconstrained, people tend to use similar techniques to indicate relations. For example, components that share visual attributes are often viewed as related in content, role or membership. Small differences

in the values of the visual attributes may imply relationship of order or sequence. Structures like lists and piles show that two or more resources share the same content or belong to the same class, and composites indicate that resources, however dissimilar, are related. Position in the workspace expresses a degree of uniqueness or importance among the components. Objects in the top left portion of the space are usually the most important or the most relevant to the user's current activity [Francisco-Revilla and Shipman. 2004]. Thus, the parser's results are evidence of perceived associations

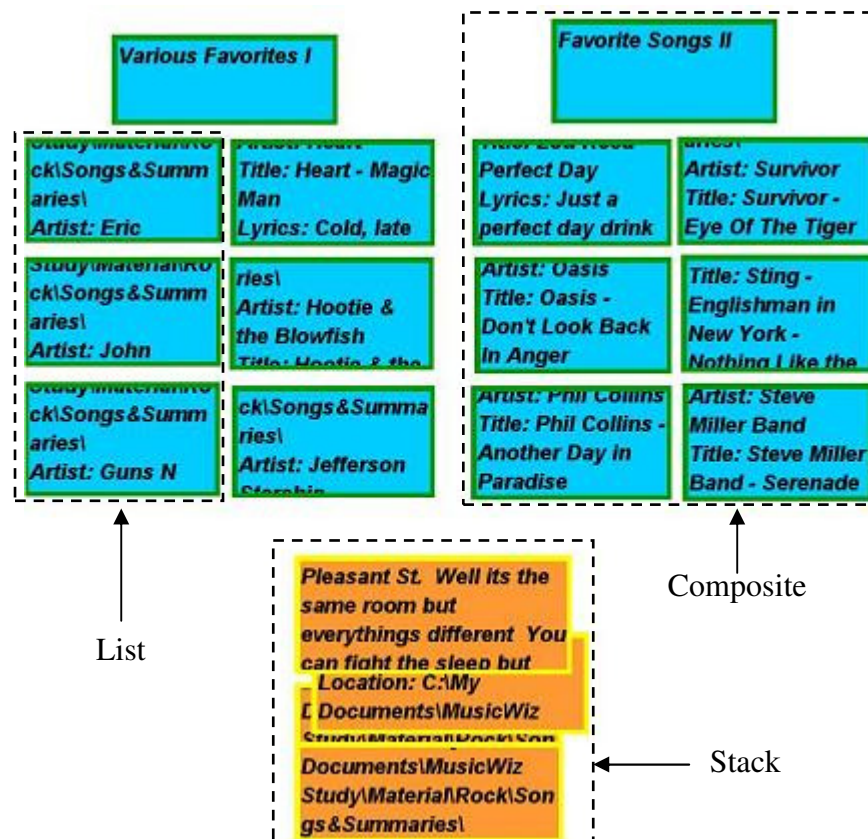


Figure 24. Example of the Structures recognized by MusicWiz's Spatial Parser

between the songs.

MusicWiz employs the same spatial parser used in VKB2 (and earlier in VIKI and VKB) [Shipman et al. 2001b; Shipman et al. 1995]. In its current configuration, it can recognize three basic types of spatial structures: lists, stacks and composites. A list consists of closely arranged, vertically or horizontally aligned objects of the same type. Objects of the same type have similar visual attributes (i.e. background and border color) and dimensions (i.e. size and shape). In a stack, objects of the same type overlap each other creating a pile. A composite describes repeating arrangement patterns of objects of different type. Figure 24 above shows examples of the three types. The song-objects in blue form four vertical lists. Each of the plain objects on top is used to label the pair of the lists below forming together a composite. Finally, the orange song-objects at the bottom form a stack.

The output of the MusicWiz parser is a forest of trees. Each tree represents a recognized spatial structure in the workspace. Song-objects part of a structure are leafs in the tree and can be in different levels. The *workspace expression module* defines the similarity of two songs ( $S_1$  and  $S_2$ ) in a tree based on the length of the path from the nearest common ancestor to the most remote leaf:

$$\text{Overall Workspace Expression Similarity } (S_1, S_2) = 1.0 /$$

$$/ (1 + \text{Level}(\text{leaf with longest path}) - \text{Level}(\text{nearest ancestor-node})).$$

### 5.2.5 Artist Module

The *artist module* assesses relatedness in music using resources already available online like human evaluations of artists' similarity and co-occurrence of artists in

playlists. Currently, MusicWiz uses the results from the research of Dan Ellis at Columbia University and especially the statistics about the co-occurrence of 400 popular artists in playlists from the *OpenNap* file-sharing network and the *Art of the Mix* website (<http://www.artofthemix.org/index.asp>). Other sources of information, such as the artist similarity from the *Similar Artists* lists of the *All Music Guide* (<http://www.allmusic.com>), could be added as an additional source of information.

Given this module's assessment is of the similarity of the artists, its output is used directly by the *metadata module* when comparing the artist name. When the similarity value for a specific pair of artists is not available via the *artist module*, the *metadata module* uses the proximity of the names as previously described with string matching techniques.

#### 5.2.6 *Generating and Integrating Module Results*

MusicWiz assesses the music similarity in two phases: a preprocessing phase where all the song features are extracted or downloaded (e.g. lyrics) and a comparison phase where all the features are compared to the features of the existing collection. New songs can be introduced to the system for this two-phase processing through the configuration dialog of Figure 10. Songs can be processed one by one or as a batch process by specifying the target file or folder(s) respectively in the textbox of the *Similarity Calculation* section. The kind of sound features (beat, centroid, and pitch) and the way those features should be extracted can be specified through the *Configuration Dialog Advanced* (see Figure 22), available from the *Advanced* button of the original dialog.

Once the extraction of the song attributes is complete, MusicWiz compares the new songs with those that have been already processed in previous sessions and records their similarity. MusicWiz's inference engine uses a weighted sum of these assessments for an overall similarity score. Users can adjust the default weights for the four modules (Metadata including Artist, Audio Signal, Lyrics, and Workspace Expression) based on their preferences (see upper part of the Configuration Dialog in Figure 13). While the default weights are set to provide what we perceive as reasonable assessments of overall similarity, the particular notion of similarity that matters depends on the user and task. MusicWiz's notion of the current task comes from expression in the workspace.

## 6. MUSICWIZ MUSIC SUMMARIZATION

### 6.1 Introduction

People use summaries to concisely describe or highlight the major points of the genuine object. In text for example, the authors of a scientific paper summarize the key points of their presentation in an abstract, a paragraph briefly describing the topic and their achievements or ideas. Accordingly, in music, vendors of CDs and mp3s (like Amazon.com and CD Universe) provide small snippets of songs to help potential customers become familiar with the contents of an album or to find songs they can only recall by melody.

Similarly, radio stations remind listeners of the top-ten hits of the week by playing the refrains of the respective songs. As the examples above indicate, a music summary (or preview) consists of one or more parts of the song that are short in duration but rich enough in information to describe and identify the total for the given current task. Such a conclusion implies that the location or the duration of those parts is not fixed for all music since their selection depends on factors like the song, the user's perception of music and the task at hand (e.g. selecting from known music or deciding whether to buy unknown music). Finding a summarization approach that takes into account all three factors requires a model of the music, a model of the user, and a model of the task. Most commercial on-line music stores preview their songs by either, the introduction of the song, a randomly selected phrase, or the (often manually selected) refrain. The simplicity of these approaches has two main problems. On one hand, there is

no guarantee that the selected phrase is sufficient for becoming familiar with or recognizing the song. On the other hand, using human resources to find the refrain for thousands of songs is costly in terms of time, effort and money. Much of the existing work in music summarization focuses on the selection of the most repeated phrase(s). When more than one phrase is selected, it is generally because the desired summary length is longer than the identified refrain and the added segment is the phrase identified as the next most frequent regardless of its similarity to the already selected refrain. As a step beyond refrain selection, this dissertation explores summaries designed to include more parts than the most salient phrase or the introduction of the song. To examine the design space for such algorithms, the author compares algorithms that compose a summary from a fixed number of components (three) but vary the selection of those components between preferring phrases that are sonically different and phrases that are repeated more often.

The following section discusses the related work in automatic summarization and a comparison of techniques.

## 6.2 Current Research

Research into techniques for the extraction of sound / music features (i.e. tempo, brightness, fundamental frequencies, bounds of phrases) is quite fertile. This work has expanded into research for developing music summaries that tend to focus on the problem of identifying musical phrases and, in particular, the refrain. Hence, the success of summarization algorithms has been typically evaluated based on how accurately they



can determine the most repeated phrase. There is a variety of approaches to identifying the refrain.

A number of algorithms [Bartsch and Wakefield. 2001, Chai and Vercoe 2003] use a pattern matching approach where the structure of the content, and more specifically the most salient phrase, is determined by comparing candidate segments (a fixed sequence of frames) with the whole song. Cooper and Foote [Cooper and Foote 2002], after the parameterization of the signal with the calculation of the Mel Frequency Cepstrum Coefficients (MFCCs), find the distance in the parameter vectors of all frame combinations and store the results in a two-dimensional self-similarity matrix. To select the segment (sequence of frames) that best represents the entire song, they calculate the similarity of each segment to the whole and choose the one with the maximum value. If the phrase is not as long as the desired summary, they add the next highest-ranking phrase(s).

Other algorithms develop more domain-specific models of the music in order to identify the most repeated phrase. Logan and Chu [Logan and Chu. 2000] use a three-step process for extracting the key phrase. After segmenting the song, they cluster the resulting segments using a modified cross-entropy or Kullback Leibler (KL) distance to infer the structure of the song and label its different parts. The key phrase is then selected based on the frequency of those labels. Lu and Zhang [Lu and Zhang 2003] use the frequency, energy and position to detect the boundaries of musical phrases by analyzing each frame's estimated tempo and computing a confidence value of the frame being a phrase boundary. Depending on the type of music (instrumental or including

vocals), Xu and Maddage [Xu et al. 2005] first extract the features that better catch the attributes of the segmented signal (e.g. MFCCs and amplitude envelope for instrumental music; linear prediction coefficients (LPCs) and derived cepstrum coefficients (LLPCs) [Rabiner and Juang 1993] for vocal music). Those features are then used for content based clustering, and the output is used for the extraction of the most representative theme. Kim et al. [Kim et al. 2006] take changes in tempo as a primary indicator for summarization. They first segment the signal based on changes in tempo and then cluster segments based on their MFCCs. Shao et al. [Shao et al. 2005] analyze a song's structure based on the rhythm and note the onset of the signal and then cluster the segments according to their melody-based (chord contours) and content-based (chord contours and vocal content) similarity. The earliest segments containing the chorus together with some directly preceding and succeeding phrases are used for the creation of the final summary. Mardirossian and Chew [Mardirossian and Chew 2006] generate music thumbnails using the sequence of the keys in time and the average time in each key to detect the most prominent melody.

Peeters et al. [Peeters, G. et al. 2002] generate a state representation of the song to discover its structural components. After discovering the potential states of the signal, they apply k-means clustering to associate each frame to one of the discovered states and a Hidden Markov Model (HMM) to identify the state sequence. The state representation is then used for the creation of the summary by choosing states and transitions according to user needs. They describe four different possible ways to generate a multi-phrase summary based on the signal analysis.

As described, most of the work on music summarization has focused on the identification of music phrases. The work presented here is complementary in that it explores the design of multi-phrase summaries once phrases have been identified.

### 6.3 Algorithms

Augmenting the refrain by compositing music phrases that are repeated in the music yet significantly different from one another can enhance the value of a summary. There are many examples where frequently occurring phrases other than the refrain are effective for recognizing a song. A highly repeated instrumental motif or a dominant verse can be as characteristic as the most salient phrase of a melody. In Lynyrd Skynyrd's "Sweet Home Alabama", for example, the introductory theme, which appears several times in the song, is almost as recognizable as the refrain itself. To explore how choices in the selection of additional phrases affects users' perceptions of the summary, the proposed summaries consist of three parts: the most salient phrase (usually the refrain) and two additional phrases. The author compares three algorithms for selecting the two additional phrases. These algorithms vary the bias between phrases that are repeated and phrases that are sonically distinct.

#### 6.3.1 *Most Salient Phrase Detection*

Phrase detection is not the focus of this work and, indeed, many of the algorithms found in related work could be used instead to identify phrase boundaries and determine repetitions. All three of the presented algorithms follow a common approach for the detection of the most salient (or key) phrase. In the preprocessing phase, the signal, after the removal of its first and last 10 seconds (that often carry non-useful information), is

segmented into fixed, non overlapping blocks of 0.75 second each. A Hamming window is applied on each block to prepare the signal for the Fast Fourier Transform (FFT), which in turn returns the frequency components of the signal. Afterwards, the algorithm calculates the MFCCs of each block as they provide a better estimation of how humans perceive frequencies.

In the next phase, groups of eight successive blocks are formed where successive groups have a 50% overlap (i.e., 4 blocks). For each group, its MFCCs are determined by taking the average of the MFCCs of its blocks. The Euclidean distance between the MFCCs of each pair of groups is then calculated and normalized. Starting with a strict (restrictive) distance threshold, clusters are computed using each group as a centroid. The largest resulting cluster is then selected. Clusters that include only contiguous segments of the music are not considered. If the threshold is too strict to generate any non-contiguous clusters, the process repeats with a more relaxed threshold.

Once the largest cluster is identified, the key-phrase is selected by identifying the block with the smallest amplitude (lowest sound level) within a range of eight blocks (6 seconds) before the starting block of each group in the cluster. The group with the smallest corresponding amplitude is selected due to the likelihood that the block is near the start of a music phrase. The start of the key-phrase is chosen to be 3 seconds prior to the selected group and the key-phrase lasts for 8 seconds.

The next subsection presents a high-level description of the three algorithms for selecting the complementary parts of the summary. This is followed by a more detailed description of how the algorithms are instantiated.

### 6.3.2 *Complementary Phrase Selections (Overview)*

The three algorithms proposed here vary the selection of the two complementary parts (segments or clusters) of the summary based on a combination of the segments' musical similarity (distance between MFCCs), the number of identified repetitions (size of cluster), and the temporal location in the musical piece. Conceptually, the first algorithm follows an approach oriented more in finding complementary parts according to their frequency of occurrence in the song. In comparison, the second algorithm increases the importance of the sonic distance in the selection process while the third algorithm places most of the emphasis on the sonic distance.

The first algorithm (Repetition Emphasis Algorithm - REA) selects the complementary phrases by placing an upper bound on the similarity between the three phrases but otherwise picks the most repeated phrases prior to and after the identified key phrase. The second algorithm (Intermediate Algorithm - IA) again selects the first complementary phrase by selecting the most repeated phrase prior to the key phrase that differs by more than a threshold. It selects the second complementary phrase to maximize the minimum of 1) the similarity between the second phrase and the refrain and 2) the similarity between the second phrase and the first selected phrase. In this way, the IA puts a higher precedence on ensuring difference between all three of the selected musical segments than it puts on the second phrase's repetition. The third algorithm (Sonic Difference Emphasis Algorithm - SDEA) goes a step further by selecting the two complementary segments that minimize the musical similarity between the three segments without considering whether the complementary phrases were repeated or not.

### 6.3.3 *Complementary Parts Selections (Details)*

The first two algorithms share their approach to selecting the first complementary part. They also steer the selection of the first and second complementary parts towards earlier and later portions of the song, respectively.

After the selection of the key-phrase from the largest cluster of blocks, the first complementary part is selected from the next largest cluster that resides, if possible, in the interval between the start of the song and the key-phrase and differs by more than a minimum threshold. The difference between the two clusters is the mean distance between the MFCCs of its groups. To be a candidate for selection of a complementary part, the mean distance must be greater than a predefined threshold and the variance of the distances must be lower than a specific limit. These thresholds reduce the likelihood that the algorithm will choose a cluster of phrases that sound very similar to the key phrase (i.e. the refrain without the voice or a variation of it). Once the next-largest cluster that meets the MFCC distance requirements has been found, the group of blocks that occurs prior to the key-phrase and is temporally most distant from the key-phrase is chosen as the first complementary part. If no groups of blocks are prior to the key-phrase, then the first complementary part is chosen to be the one closest to the end of the song (furthest from the key-phrase).

The selection of the second complementary part in the REA proceeds similarly. Again, the algorithm selects the next largest cluster with significant differences in MFCC means from both the key-phrase and the first complementary part. Once the cluster is identified, the group of blocks closest to the end of song is selected (assuming

the first complementary part was chosen from before the key-phrase, otherwise it will select the group of blocks closest to the start of the song).

In the IA, the selection of the first complementary part uses the technique described in the first algorithm. However, the selection process of the second complementary part deviates significantly except that the search is still focused on the interval between the key-phrase and the end of the song. The second complementary part is chosen as the group that maximizes the minimum of the value of  $F(i)$  in formula (1):

$$F(i) = \min (DK(i), DP(i)) \text{ where (1)}$$

$$DK(i) = L (V(\text{group}_i), V(\text{key-phrase})) \text{ (2)}$$

$$DP(i) = L (V(\text{group}_i), V(\text{first complementary part})) \text{ (3)}$$

where  $L (V1, V2)$  is the Euclidean distance of the MFCC vectors  $V1, V2$  and  $i$  the number of the groups for the portion of the song being examined.

The SDEA differs substantially. After the extraction of the key-phrase, the ten least similar groups of blocks (in terms of MFCC vector similarity) to the key-phrase group of blocks are used as candidates for the selection of the phrases. From the ten candidates, the pair with the minimum similarity (maximum Euclidean distance) is used for the extraction of the two complementary parts. Thus, this algorithm places greater emphasis on sonic difference.

#### 6.3.4 *Summary Creation*

To create the final summary, a eight-second slice is taken from the key phrase and a six-second slice from each of the two complementary parts (total twenty seconds) and ordered temporally. A one-second silence is introduced between the segments to

diminish the effects of the abrupt switches. Fading in and out is not currently used since it “steals” potentially valuable time from the summary.

However, the evaluation indicated that smoothing the transitions between phrases is important to users so cross-fades will be part of author’s future efforts. The final summary has length twenty-two seconds, which is comparable to many of the commercial summaries, found.

#### 6.4 Evaluation Design

An experimental study was designed to evaluate and compare the three summarization approaches and to test their performance over a widely used technique. The study was conducted in the Center for the Study of Digital Libraries at Texas A&M University. Fifteen participants over 18 years old, mainly students, were recruited to take part including 12 men and 3 women. The majority (67%) had some kind of music education and more than half of them (67%) had a personal music collection of at least 50 songs (8 participants had more than 200 songs).

Participants were asked to listen carefully to the summaries of twenty popular rock and pop songs and choose the summary that best represented each song. In contrast to the study conducted by Ong [Ong, B. 2006], the evaluation criteria used in this study were focused on user preference and summary completeness and not on metrics measuring the ability of subjects to assign song titles. There were four 22-second summaries per song, three generated with our algorithms and one that was merely the first 22 seconds of the song.



Participants answered a series of multiple-choice questions about the quality of the selected summary and their familiarity with the song before proceeding to the next one. The summaries, as well as the songs (in their full version), were accessible through a web-based interface.

Participants were able to navigate through the songs and listen to the summaries as many times they wanted. There was no time limit for the completion of the task. The order in which the songs and the summaries were presented to the participants was balanced across participants. Demographic data about the participants was collected via a pre-task questionnaire. Post-task, semi-structured interviews were used to gather information about the participants' perceptions of the task, their experience with the algorithms and their ideas for future improvements.

## 6.5 Evaluation Results

To get an idea of what users really appreciate in a music summary they were asked to name (pre-task questionnaire) the parts or features of songs they consider

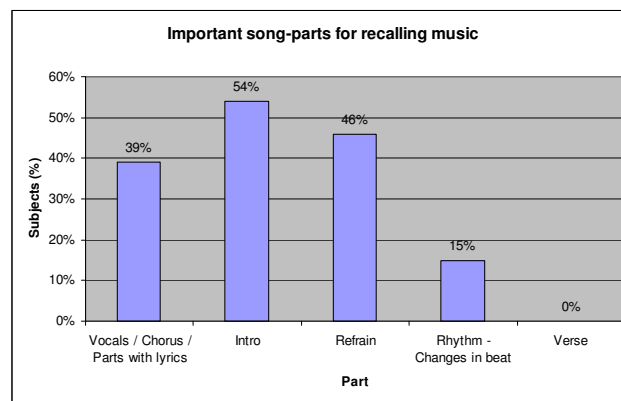


Figure 25. Important Parts for Recalling Music

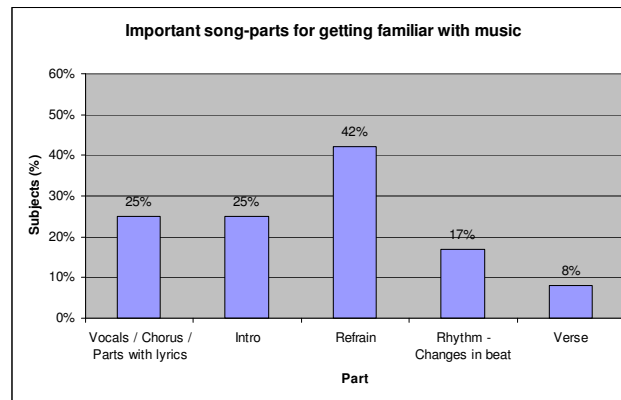


Figure 26. Important Parts for Familiarizing with Music

fundamental for becoming familiar with and recalling music.

The results confirmed that both the introduction and the refrain are believed to have an important role in the process of understanding and recognizing music (see Figures 25 & 26). However, there was a distinction between the two cases. The introduction of the song was indicated most important for remembering (although with small difference from the refrain that was second) while the refrain was ranked best for

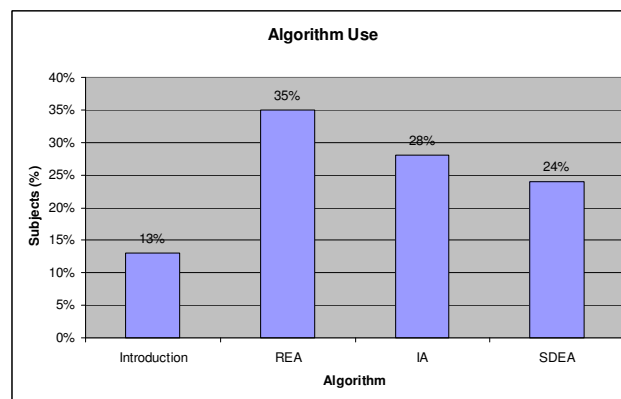


Figure 27. Users' Algorithm Choice

becoming acquainted with the music. The higher score of the introduction and the vocals / chorus / lyrics in recalling music matches the fact that a few words or notes can be sufficient for identifying a song we know but provide little information for a song we do not know. Finally, other musical parts like bridges and verses scored very low in the preference ranking or were not mentioned at all by the participants.

Figure 27 shows the distribution of participants' selections for their favored summary. Participants chose the introduction summary in only 13% of the cases and the REA, the most popular, in 35% of the cases. Analysis of the data shows the difference in the selection of the four algorithms was statistically significant (F-test,  $P=0.0013$ ,  $\alpha=0.05$ ). Table 2 presents the results from the pair-wise comparison of the algorithms with the Tukey HSD test. The numbers show a statistically significant difference between the introduction-based algorithm and the REA and IA ( $P=0.001$  and  $P=0.041$  respectively,  $\alpha=0.05$ ).

Table 2. Pair-wise Comparison of the Four Algorithms

(I) Algorithm	(J) Algorithm	(I-J) Mean Diff.	Std. Error	Sig.
Introduction	REA	-4.4286	1.03067	<b>.001</b>
	IA	-2.8571	1.03067	<b>.041</b>
	SDEA	-2.1429	1.03067	.178
REA	Introduction	4.4286	1.03067	<b>.001</b>
	IA	1.5714	1.03067	.433
	SDEA	2.2857	1.03067	.136
IA	Introduction	2.8571	1.03067	<b>.041</b>
	REA	-1.5714	1.03067	.433
	SDEA	.7143	1.03067	.899
SDEA	Introduction	2.1429	1.03067	.178
	REA	-2.2857	1.03067	.136
	IA	-.7143	1.03067	.899

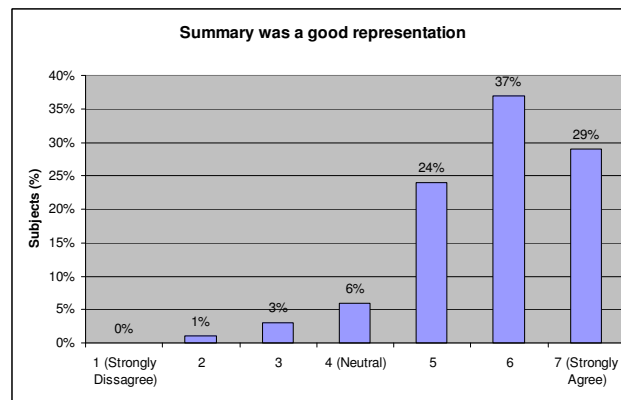


Figure 28. Evaluation of Summaries' Performance

While there was no statistically significant difference between the three multi-phrase algorithms, the trends in the data show that identifying repeated phrases is likely to add value to the resulting multi-phrase summary.

The correlation between the algorithm choice and how good the summaries were (see Figure 28) wasn't statistically significant. However, the numbers look promising considering that 13 of the participants evaluated their choice as at least a good representation of the song, and this choice was one of our algorithms in 87% of the songs. However, the most interesting point about the weakness of the song introduction as summary came out from the post-task interviews. The analysis showed that, while a respective number of the participants (10) listen to the song previews provided by the on-line music stores, only 3 of them are confident that these previews describe the songs sufficiently. Since most online stores preview music using a single contiguous snippet, this indicates a need for an alternative to current summaries.

Participants reported knowing 71% of the songs well, not knowing 16% of the songs, and having limited knowledge of 13% of the songs. Analysis of the data showed that there is no statistically significant correlation between the choice of the algorithm and how familiar the participants were with the music. Table 3 shows the number of times participants selected each algorithm based on their knowledge of the song. The proposed techniques are superior in effectiveness over the traditional introduction-based approach no matter whether the user is a customer browsing a new album or a person trying to retrieve an already known mp3 from a personal collection.

Strongly related to participants' knowledge of the songs is how recently they had heard them. Participants had listened on average to about 59% of the songs within a year (see Figure 29). However, the statistics again did not show a significant correlation with the algorithm choice. One of the concerns in designing the proposed summarization techniques was that the segments in each summary would be too short to sufficiently describe the section of the song that had been extracted. A music phrase (especially in classical music) can have duration much longer than the six seconds selected as the length for complementary parts. However, for this collection of pop and rock songs, the

Table 3. Algorithm Selection and Familiarity with Music

Algorithm \ Knowledge of Songs	I know it well	I don't know it well	I haven't heard the song
<b>Introduction</b>	32 (16%)	4 (11%)	1 (2%)
<b>REA</b>	66 (33%)	10 (28%)	23 (54%)
<b>IA</b>	53 (26%)	14 (39%)	10 (23%)
<b>SDEA</b>	50 (25%)	8 (22%)	9 (21%)

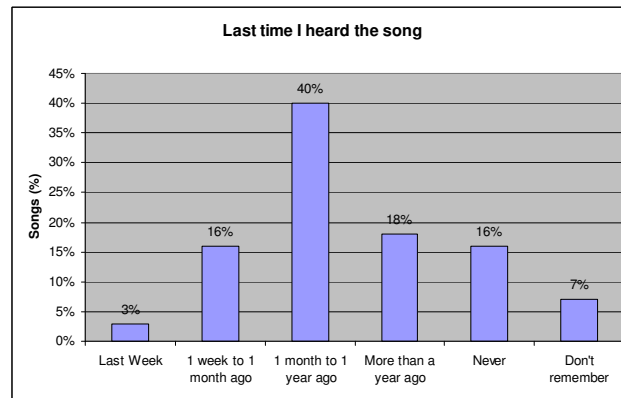


Figure 29. Participants' Last Time of Listening to the Song

results showed that only 20% of the selected summaries were considered “too short” while 53% evaluated were considered “good” and 27% were viewed as “too long”.

## 6.6 Discussion

Three algorithms for creating multi-phrase music summaries were developed and evaluated as part of this dissertation. The design of the algorithms reflects a range of approaches that vary between emphasizing the selection of repeated phrases and the selection of sonically different phrases. The study showed that participants believed that the multi-phrase summaries better represented the song than the introduction to the song. While the difference between the three algorithms was not significant, the results indicate a likely preference for algorithms that emphasize the selection of repeated phrases, at least in the genre of pop and rock where the structural components of the melody are more standardized and identifiable.

There are several potential improvements to the above algorithms that could be considered for future research. One of the complaints participants had during the task

was that the switch from part to part in the summaries was too abrupt and hence distracting or even annoying. Use of phrase bounds detection for selecting the start of phrases could help as could the use of fade-in and fade-out effects. Based on participants' feedback about which parts / features of the songs are important, it would be interesting to examine if the integration of the introduction in the proposed summaries can improve or accelerate the process of becoming familiar with new music. A comparison of the best of the presented techniques with summaries containing only the most salient phrase of the song would be a better comparison of multi-phrase summarization to the approaches found in the research literature.

Finally, the current summarization approach works well for pop and rock but not for jazz and classical music. Future work could expand the summarization approach to perform better in such genres, where identification of the various themes and important components is more challenging.

## 7. MUSICWIZ EVALUATION

There are two central hypotheses in the design of MusicWiz: that a freeform workspace and suggestions based on a multi-faceted similarity metric will be valuable for collection management and use. A comparative study examining the effects of the workspace and the suggestion took place to evaluate these hypotheses.

### 7.1 Evaluation Design

The study took place in the same place as the preliminary study. The twenty volunteers, mainly white (15 out of 20) males (16 out of 20) under 36 year of old (18 out of 20), had in their majority (14 out of 20) some kind of formal music education (Figure 30, left). Most of them (15 out of 20) had a personal mp3 collection of at least 50 songs (Figure 30, right) of multiple genres (17 out of 20, See Table 4) organized (10 out of 20) without the use of specific software for management / browsing (12 out of 20). Only five reported organizing / listening to their songs four or more times per week and just

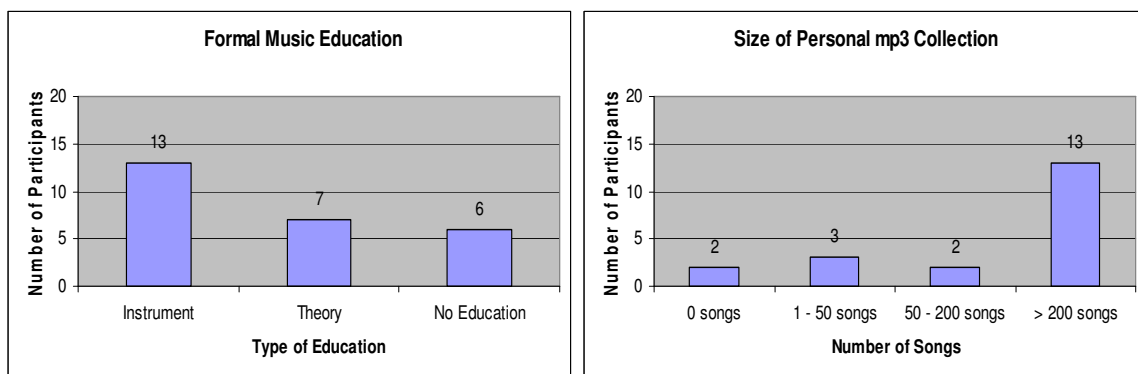


Figure 30. Music Education Levels and Collection Size



Table 4. Participants' Preference on Genre

Rock	Classic	Pop	Folk / Ethnic	Jazz	Blues	Electr.	Salsa	Rap	Altern.
<b>13</b>	12	9	9	6	4	4	3	3	3
<b>(65%)</b>	(60%)	(45%)	(45%)	(30%)	(20%)	(20%)	(15%)	(15%)	(15%)

two spending more than 10 minutes in organizing per sitting. However, sixteen listen to their music for at least a half an hour per sitting (See Figure 31). From those that claimed they use the metadata / statistics of use / rating filtering-support of their media players (or managers) for creating playlists (6 out of 20), four rated their satisfaction from the system-returned songs as a five or six in a nine-point Likert-scale (ranging from 1 – “I need to search several times” to 9 – “Very close to what I want to listen to”). The other two valued their satisfaction as a seven and eight in the same scale. Finally, only participants indicated reusing their playlists more than five times while thirteen reported not rating their songs.

Participants were given a collection of fifty classic rock songs and asked to complete three tasks: one requiring classification of the music and two involving searching and similarity assessment. In the first task, songs had to be organized into sub-collections according to participants' own categorization scheme. There was no restriction in the number, the type or the content of the sub-collections that had to be created. In the second task, participants had to form three playlists, twenty minutes long each, based on three different moods or occasions of their choice using songs from their

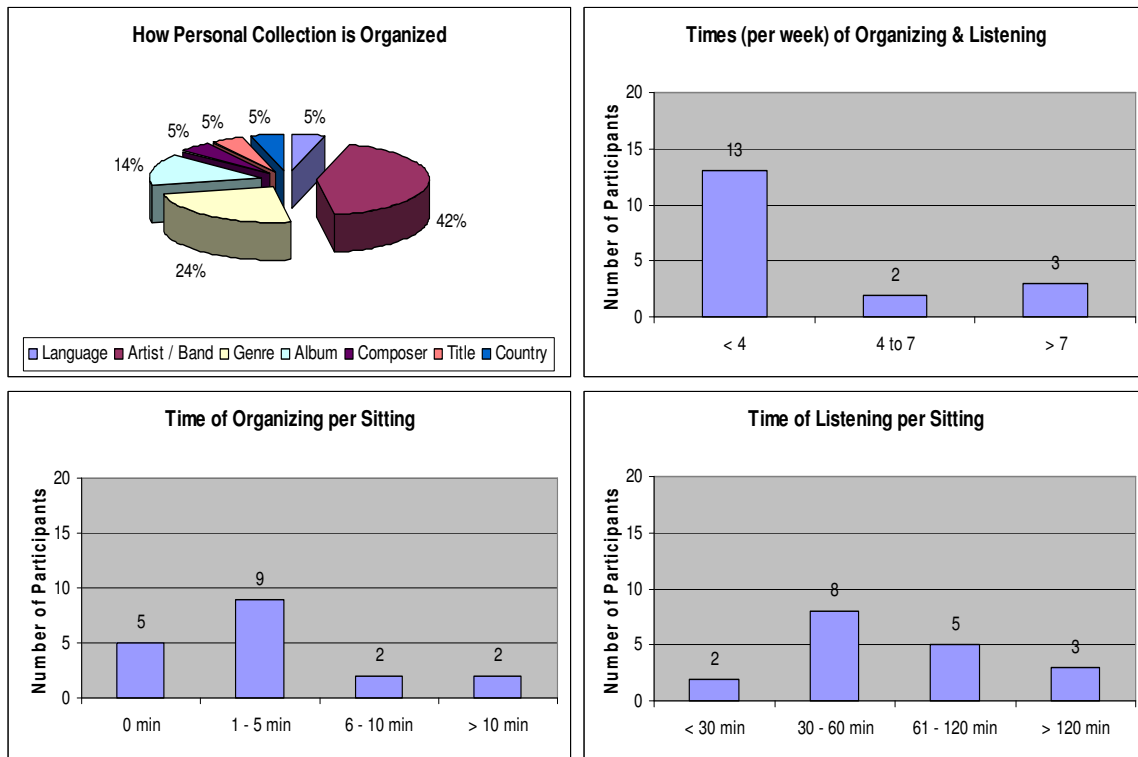


Figure 31. Music Collection Structure - Organization and Listening Habits

sub-collections. In the third task, participants had to form three playlists, six songs long each, using their sub-collections, but this time the content of each playlist had to be similar (or somehow related) to a specific song (not from the fifty of the original collection) we were providing them as a seed (example). Participants had unlimited time to complete the tasks and playback access to all of the songs.

To assess the contribution of MusicWiz's workspace and similarity suggestions, participants were divided (equally and randomly) into four groups of system use. Participants in the first group (no workspace / no suggestions) had to complete all three tasks using only the playback and search functionality of the system and Windows

Explorer folders to form the sub-collections and playlists. The participants of the second group (no workspace / with suggestions), were allowed to use the features the participants of the first group could use and additionally similarity suggestions provided by the system (e.g. suggestions from the related songs feature or the automatically created playlists). In the third group (with workspace / no suggestions), participants had to perform the tasks using the features available in the first group with the only difference that they had to use the MusicWiz workspace to create the collections and the playlists. Finally, the participants of the last group (with workspace / with suggestions) had all the features of the system available including the workspace. Table 5 summarizes the four group configurations.

The use of Windows Explorer folders for the “No Workspace” conditions rather than a music management application (like iTunes) was based on a combination of evidence that many people use the file system to manage their collections and that it would be the most familiar interface across participants. The demographic data found that only 25% of the participants in the preliminary study and 30% of the participants in the current study used specialized software for organizing their music collection.

Table 5. MusicWiz’s Configurations for Study Groups

<b>Configuration</b>	<b>No Suggestions</b>	<b>Suggestions</b>
<b>No Workspace</b>	Group 1	Group 2
<b>Workspace</b>	Group 3	Group 4

## 7.2 Evaluation Results

Results from the study include quantitative data about participant activity (e.g. the time taken for tasks), participant assessments from seven-point Likert-scale responses (ranging from 1 – “I strongly disagree” to 7 – “I strongly agree”), and open ended comments.

### 7.2.1 Task One: Classification of Music

The average time taken to organize the music collection varied across the different configurations with the average completion time of task one for Group 1 being 46.2 minutes (longest of the four groups,  $s = 11.05$ ) while the respective time for Group 3 was just 28 minutes (shortest of the groups,  $s = 13.02$ ). This difference approaches statistical significance ( $\alpha = 0.1$ ,  $p\text{-value} = 0.0625$ ) according to the Wilcoxon test. The average time for Group 2 of 44 minutes ( $s = 15.48$ ), was close to Group 1 and the

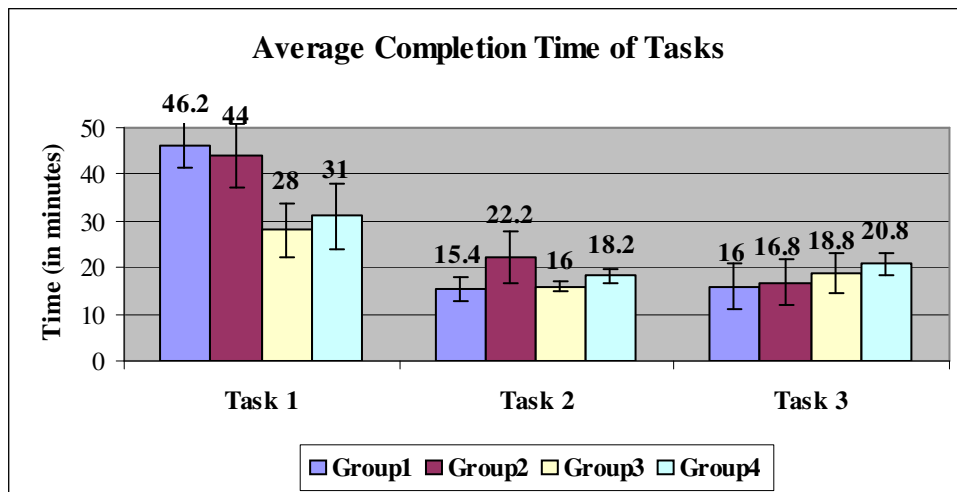


Figure 32. The Average Completion Times of the Participants in the Three Tasks

completion time of 31 minutes ( $s = 16.02$ ) for Group 4 participants was similar to that for Group 3 (see Figure 32). The Anova and Kruskal-Wallis tests did not reveal any statistically significant differences in the average completion times of the four groups. However, the results indicate that the workspace made the organization task more efficient while the suggestions neither helped nor hindered the time to generate an organization.

These time results are supported by participants' assessments on the quality of support they were provided by the system. In the statement "I had enough support to effortlessly / quickly organize the songs the way I wanted", the average rating of participants in Group 1 was 4.4 ( $s = 1.52$ ). That was the lowest among the four groups. The participants of Group 3 rated the support they had as a 5.6 ( $s = 0.89$ ). The participants of Group 4 appeared to be the most satisfied of all with an average rate of 6.2 ( $s = 0.83$ ) while their counterparts in Group 2 answered with a 5.4 ( $s = 1.95$ ), which was the second lowest score. Participants of Group 3 and Group 4 had quicker access to music provided by the song previews available in the workspace. This interpretation is supported by the comments of several participants about the significance of the song-previews in the fast assessment of the music.

Participants in Group 1 were also those that were most negative regarding the statement "I enjoyed doing this task" (avg. 5.4,  $s = 1.34$ ). The average ratings for the other three groups (starting from Group 1) were 5.8 ( $s = 1.64$ ), 6.4 ( $s = 0.55$ ) and 6 ( $s = 1$ ). Assuming that participants are more likely to enjoy a task where they are provided with assistance, the results look reasonable and consistent with how participants rated

the support provided by the system. The most unexpected result came from the statement “it will be easy for someone else to understand the way I organized the songs”. Consistent with prior statements, participants of Group 4 were the most positive (5.8 avg.,  $s = 1.1$ ). Surprisingly, the most reluctant were those in Group 3 (4.2 avg.,  $s = 1.64$ ). Participants in Group 1 and Group 2 agreed on a 5.4 (avg.,  $s = 0.55$  and  $1.52$  respectively). This indicates an interaction between the workspace features and the suggestion features. One interpretation is that the MusicWiz workspace, supporting free-

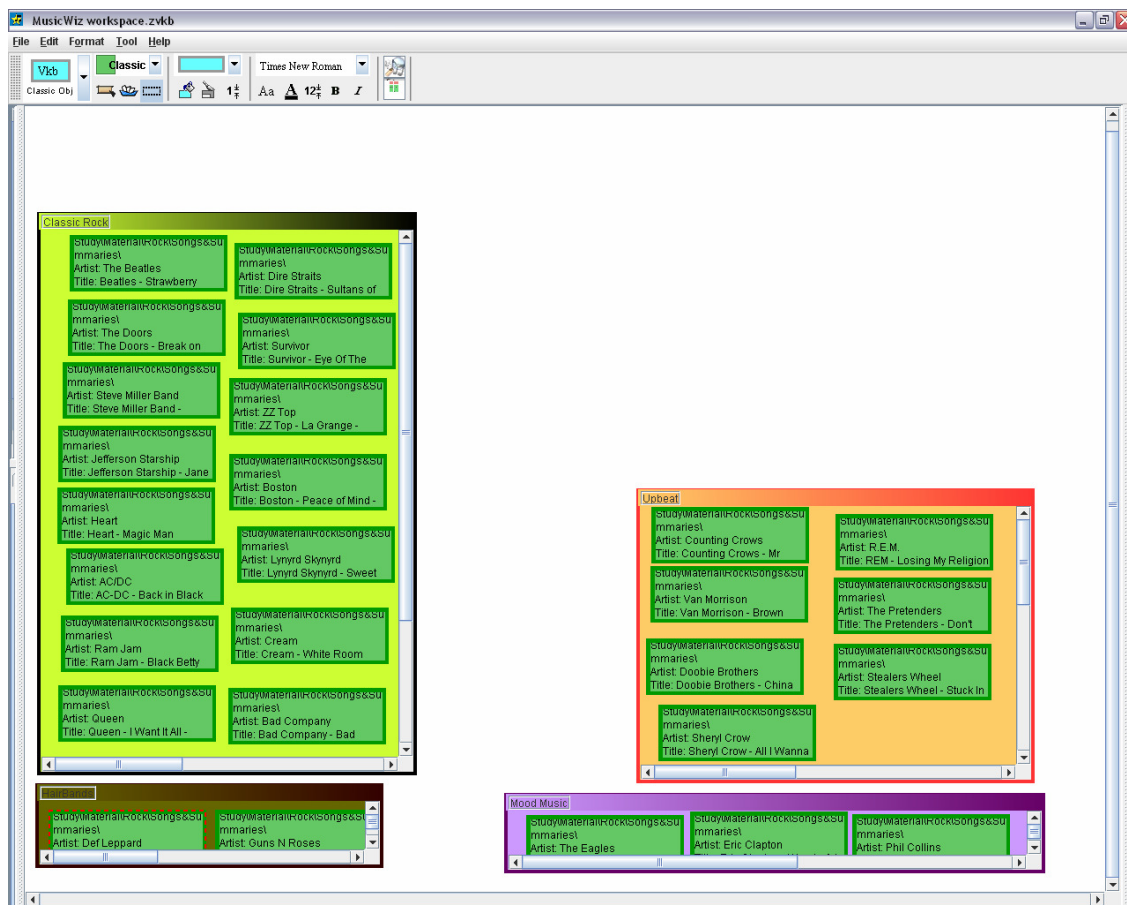


Figure 33. Organization Based on Music Genre and Dynamics (Group Four)

form expression, encourages music association in an implicit way (i.e. based on how music sounds or is perceived) rather than relying on the explicit information to provide consistency and repeatability. Without the system suggestions, Group 3 participants organized the songs in a way that was making sense to them but not necessarily sense to anyone else. Following the same logic, the confidence of the participants in Group 4 may derive from the confirmation and support for the initial organization provided by system suggestions.

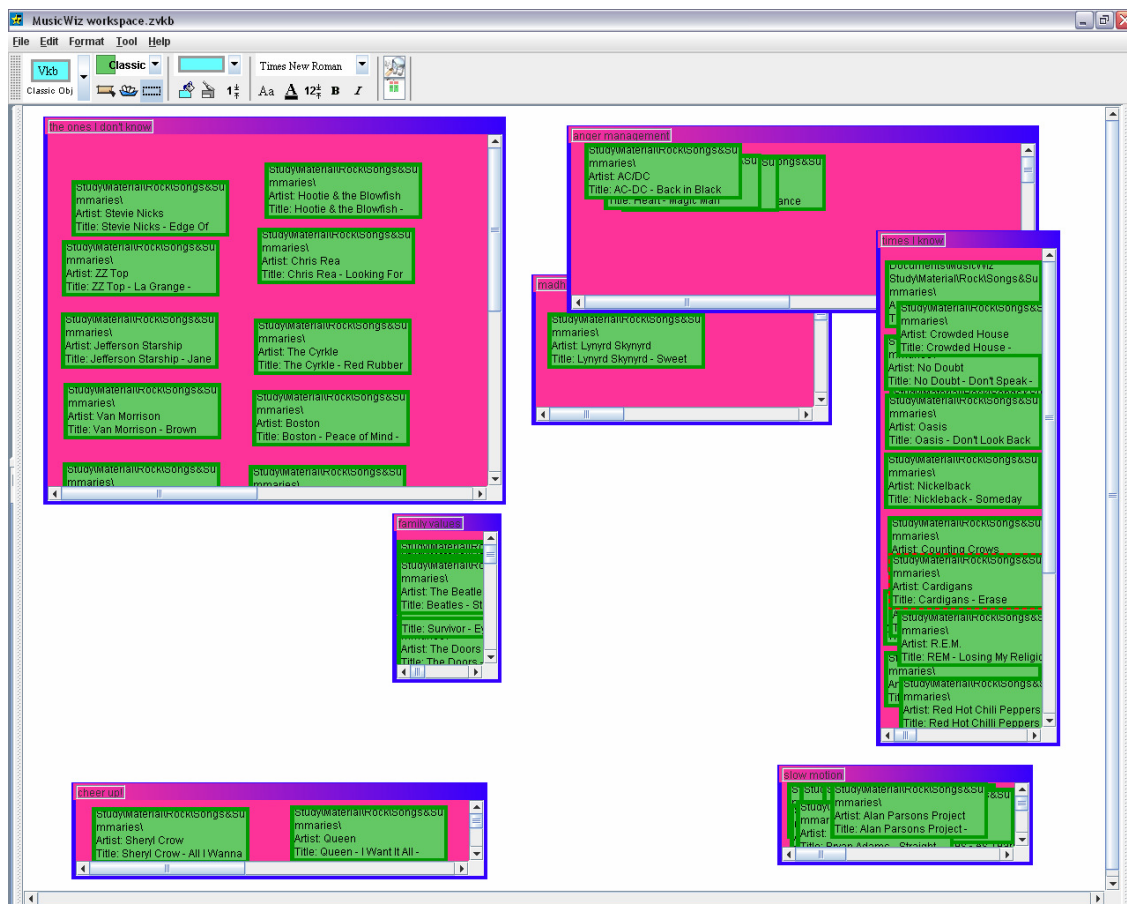


Figure 34. Organization Based on Music Knowledge, Concept of Listening and Music Dynamics (Group Three)

Figures 33 and 34 show the organizations created by two participants in MusicWiz during task one. In the organization shown in Figure 33, a combination of implicit and explicit attributes of music has been used to assign songs into four main collections. The collections on workspace's left side (descriptors "Classic Rock" and "Hair Bands") are typical examples of music classification based on metadata information (genre specifically). The collections on the right however (descriptors "Upbeat" and "Mood Music"), show a less conventional grouping based on music

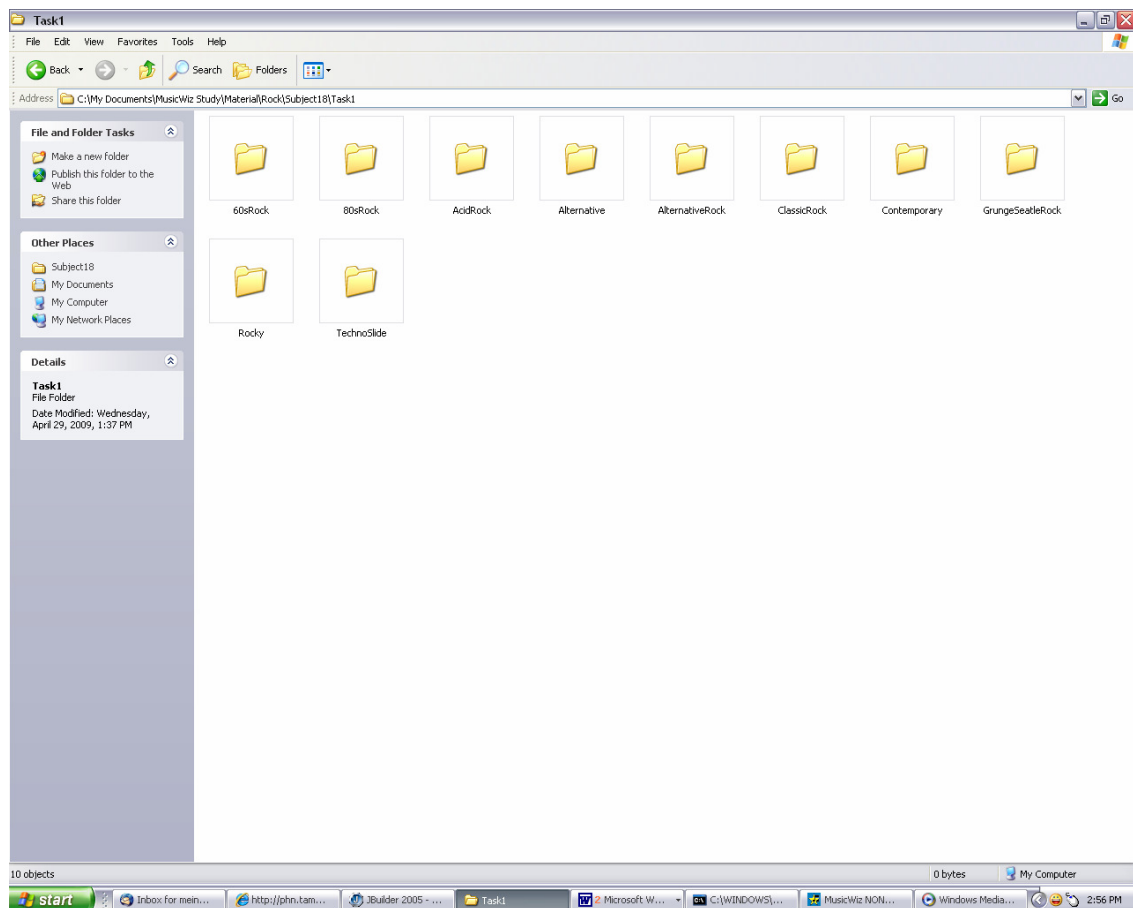


Figure 35. Genre-based Classification in Windows Folders (Group Two)



dynamics. In the organization in Figure 34, the two dominant criteria for associating the music are the knowledge of the songs and the concept of listening. The participant has created the collection “songs that I don’t know” to isolate all the songs he was not familiar with. The rest of the songs have been assigned into collections according to their sound / harmony dynamics (e.g. “cheer up” and “slow motion”) and the occasion / mood the user would like to listen them (e.g. “cheer up!” and “anger management”). Figure 35 above shows the organization created by one of the participants of group two. The songs have been assigned into folders based on traditional metadata information and more specifically their rock subgenre (e.g. “Acid Rock”, “Alternative Rock”, “80s Rock” and “Grunge Seattle Rock”).

### 7.2.2 *Task Two and Three: Playlist Creation by Concept and by Example*

The time to complete the playlist creation tasks showed no significant differences across the conditions. The Likert responses were fairly similar for playlist creation tasks as they were for the organization task. When rating the statement “I had enough support to effortlessly / quickly browse and select the songs” for their playlists, the participants in Group 2 and Group 4 were the most satisfied with rates from 6.2 and over, followed closely by the participants in Group 3 (5.8 and 5.6 avg.,  $s = 0.84$  and  $1.95$  in tasks two and three respectively). The participants in Group 1 were barely positive when evaluating system support with 4.8 and 4.4 average ( $s = 1.64$  and  $1.52$ ) on the two tasks. The comparison of the average scores of the groups that had the system suggestions and

Table 6. Average of Seven-point Likert-scale Ratings for Playlist Creation – Higher Values Are More Positive

Statement	Task	Group 1	Group 2	Group 3	Group 4
Support for quick selection	Two	4.8	<b>6.2</b>	5.8	<b>6.2</b>
	Three	4.4	<b>6.8</b>	5.6	6.2
Support for finding	Two	4.8	6.0	5.4	<b>6.8</b>
	Three	4.6	<b>6.4</b>	6.0	<b>6.4</b>
Enjoyed doing task	Two	5.2	6.0	5.8	<b>6.4</b>
	Three	5.2	5.8	6.4	<b>6.6</b>

those that had not the feature revealed statistically significant differences in task three ( $t(9) = -2.42$ , two-tail  $p = 0.031$ ). Table 6 provides a summary of the averages for all tasks and groups – the most positive assessment for each statement/task is shown in bold.

When asked about the statement “I had enough support to browse and find the songs I was interested in”, the participants of Group 1 provided again the least positive responses (4.8 and 4.6 avg.,  $s = 1.48$  and  $1.52$  in tasks two and three respectively). The participants in Group 4 strongly agreed on the sufficiency of their system (6.8 and 6.4 avg.,  $s = 0.45$  and  $0.89$ ) indicating the effectiveness of the MusicWiz in capturing user preferences and identifying music of interest. Group 2 participants were almost as positive (6 and 6.4 avg.,  $s = 0.71$  and  $0.55$ ). Without suggestions but with all the other

system features enabled (searching, tree-view of the collection etc.), the participants in Group 3 rated the support they had as a 5.4 and 6 ( $s = 1.14$  and  $1.41$ ) in the two playlist-creation tasks. The average score difference between the groups with the system suggestions available and those without was found to be statistically significant in task two ( $t(9) = -2.81$ , two-tail  $p = 0.014$ ).

The enjoyment factor proved to be higher for Group 4 on both playlist creation tasks (6.4 and 6.6 avg.,  $s = 0.89$  and  $0.55$  for tasks two and three respectively) than the

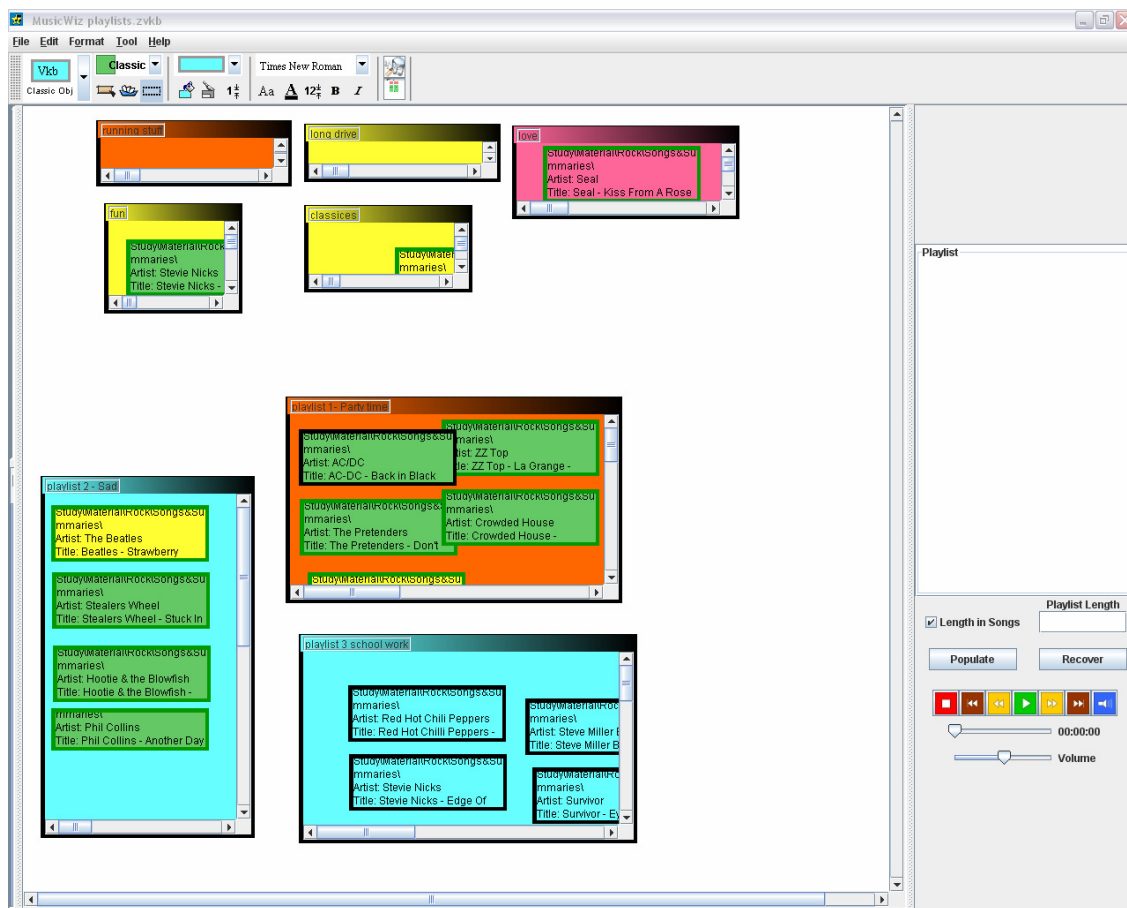


Figure 36. Preference and Concept-based Playlist Creation (Group Four)

participants in any other group (5.2 and 5.2 avg.,  $s = 1.64$  and  $1.64$  for Group 1, 6 and 5.8 avg.,  $s = 0.71$  and  $0.84$  for Group 2, and 5.8 and 6.4 avg.,  $s = 0.84$  and  $0.55$  for Group 3 in the two tasks). The difference in the replies of the groups with and without the workspace was found to be statistically significant in task three ( $t(9) = -2.30$ , two-tail  $p = 0.04$ ). Overall, these results imply that suggestions are more important for supporting playlist creation than the workspace, although the workspace enhanced participants' satisfaction and enjoyment as well as their perceptions of support.

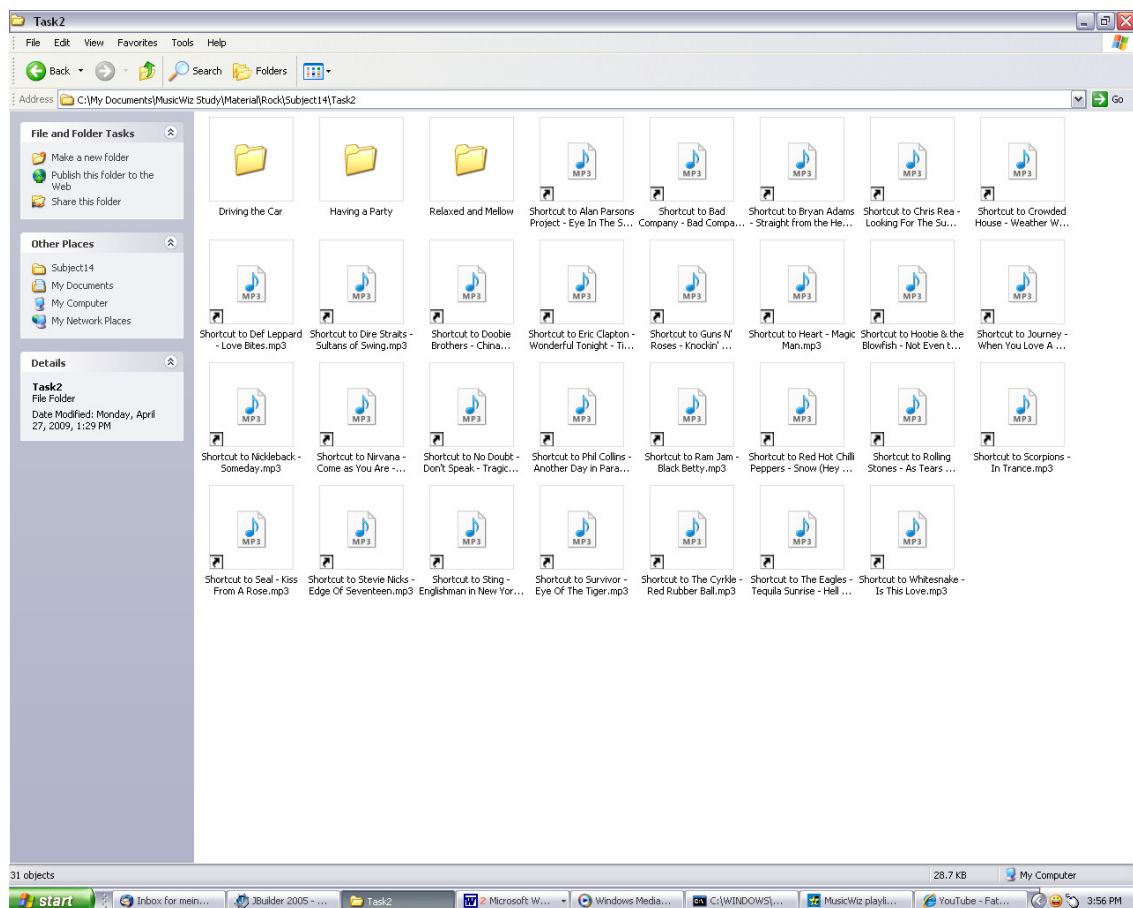


Figure 37. Concept and Event-based Playlist Creation (Group Two)

Figure 36 presents the three playlists one of the Group 1 participants created for task two. Two of the playlists have been formed according to a specific occasion / activity (“Party Time” and “School Work”) while the third one based on a mood status (“Sad”). Notice the use of the various colors for the backgrounds and borders of the songs and their collections. Figure 37 shows the resulting playlists in the same task for a Group Two participant. As in the previous example, two of the playlists have been formed according to the concept of a specific event (“Driving the Car” and “Having a

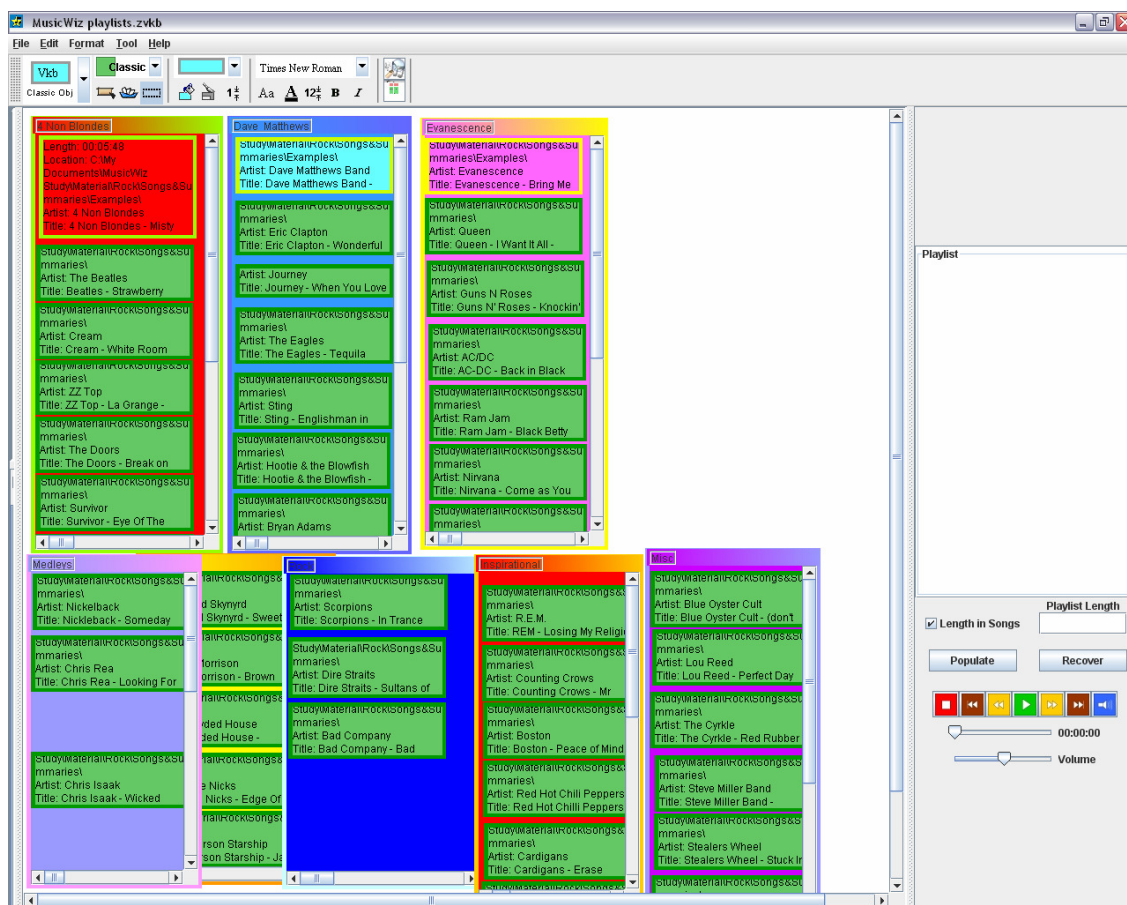


Figure 38. Playlist Creation by Example (Group Three)

Party”) while the third one to express a specific mood (“Relaxed and Mellow”). As participants were instructed to build their playlists according to personal moods or occasions, not much variety was expected to be seen in the themes of the playlists across the different groups.

Figure 38 shows the playlists (top of the workspace) created by one of the Group 3 participants in task three. This participant used unique colors for distinguishing the example-songs (i.e. given seed for the playlist) from the retrieved / selected ones on each playlist.

## 7.3 Discussion

### 7.3.1 *Organization Tactics*

Regardless of the group and hence the functionality that was available in each case, participants created organizations and playlists using a variety of criteria and attributes including metadata, melody and sound dynamics, concepts or occasions of listening and their preferences. The organizational structures created by the participants indicate what aspects of music they consider important. Examining the labels of the structures in the various conditions may indicate whether and how the tools in the different conditions are affecting these aspects. Table 7 summarizes the labels assigned to the collections and playlists participants created during task one and two. A color code scheme has been applied to highlight the different kinds of collections participants created. Purple indicates metadata based classification, yellow music-content driven classification, green preference / knowledge oriented classification and red concept /

occasion / event influenced classification. Table 8 shows the distribution (in number of collections) of the different criteria and attributes across the four groups.

Table 7. Collection and Playlist Labels Created at Task One and Two

Group	Task One - Organizations	Task Two - Playlists
One	“Comfortable and Soft”, “Loud and Quick”, “Sad”, “Quick but not Noisy”	“Slow”, “Sad”, “Loud”
	“Don’t Like”, “Favorite”, “Heavy”, “Soft”, “SoSo”	“Driving”, “Party”, “Smooth”
	“Level 1”, “Level 2”, “Level 3”, “Level 4”, “Level 5” (the higher the level the faster / harder the music is according to participant’s comments)	“Party”, “Summer Holiday”, “Road Trip”
	“Rock”, “Pop”, “Chill out”	“Car”, “Party”, “Relaxing”
	“Drive”, “Work”, “Will Not Listen”	“Relaxing Party”, “Travel”, “Dance Party”
Two	“Hard”, “Calm”, “Slow”, “Uplifting”	“Bad Mood”, “Calm Mood”, “Cuddling Mood”
	“Mellow”, “Rock”, “Energetic”	“Energetic”, “Driving”, “Study”
	“British Invasion”, “Driving Beats”, “Hair Metal”, “Hard Rock”, “Laid Back”, “Melancholy Music”, “Swinging Tunes”	“Driving the Car”, “Having a Party”, “Relaxed and Mellow”
	“60sRock”, “80sRock”, “Acid Rock”, “Alternative”, “Alternative Rock”, “Classic Rock”, “Contemporary”, “Grunge Seattle Rock”, “Rocky”, “Techno Slide”	“Working out”, “Long car ride”, “Coffee shop listening”
	“Pop” (“Country”, “Edgy”, “Smooth”), “Rock”	“Party”, “Driving”, “Romantic”
Three	“80s”, “Classics”, “Recent”, “Ballads”	“Ballads/Soft Rock”, “A bit upbeat”, “Classic Rock”
	“Hard”, “Soft”	“Soft / general purpose”, “Slow dance”, “Hard / workout”

Table 7. Continued

Group	Task One - Organizations	Task Two - Playlists
Three	“Classic”, “Hard/Fast”, “Mellow/Ballad”, “Grunge/Bitchy”	“Driving”, “Chillin”, “Company”
	“The ones I don’t know”, “Family Values”, “Cheer up!”, “Slow Motion”, “Anger Management”, “Mad Hatters”, “Times I know”	“Meeting with friends”, “down-mood songs”, “oldies party”
	“Funky”, “Easy Going”, “90’s ish”, “Heavyish”, “60’s 70’s ish”, “Stuff I don’t listen to”	“Pool Hall”, “Running”, “Chill”
Four	“80s”, “Soft”, “Heavy”, “Strawberry”	“Driving”, “Working”, “Dinner with friends”
	“Old”, “Loud”, “Mix”	“Angry”, “Late”, “Happy”
	“Pop”, “Soft”, “Hard Rock”, “Slow”	“Love”, “Easygoing”, “Party”
	“Running stuff”, “Long Drive”, “Love”, “Fun”, “Classics”, “Rock”	“Party Time”, “Sad”, “School Work”
	“Classic Rock”, “Hair Bands”, “Upbeat”, “Mood Music”	“Night Driving”, “Hard rock”, “Upbeat”

There was a wide variety of tactics used for the first task of organizing the collection. From the total of 92 collections resulted from task one, about 33 (35.9%) formed based on metadata information – mainly subgenre category. That was an unsurprising result as the study setup had been designed such that to intentionally discourage the use of metadata for classification. It is not random that all the songs were from the same general genre, a wide range of release years, and non-overlapping albums,



Table 8. Number of Collections per Group and Type in Task One

Criteria Used in Task One	Group One	Group Two	Group Three	Group Four	Total
Metadata	3	14	8	8	33 (35.9%)
Music / Sound Content	12	9	6	8	35 (38%)
Preference / Knowledge	3	0	3	0	6 (6.6%)
Concept / Occasion / Event	2	5	6	5	18 (19.5%)
<b>Total</b>	20	28	23	21	92

bands and artists making the association based on those attributes hard. The study was intended to evaluate the support of the proposed features in improving the music management experience and not to verify if people would rely less on explicit attributes when organizing music in MusicWiz (that was the focus of the preliminary study in section 4.2).

A look at the numbers of table 8 reveals that the participants in Group 2 were the ones that created most of the metadata-based collections (14 out of the 33 collections). On the other side, the participants of Group 1 proved to be the least tempted to use metadata for their classifications (3 out of the 33 collections) and at the same time the ones that created the highest number of collections based on the music dynamics (12 out of the 35 collections). The two extremes in the metadata use occur between groups

where participants had to work in the same system environment (the Windows Explorer). It is not clear how the availability of the similarity suggestions in Group 2 - the capability not available to Group 1 participants - increased the use of metadata. A possible interpretation is that, since MusicWiz is able to generate suggestions based on metadata values, it was much easier and straightforward for the participants of Group 2 to create metadata-based classifications than their counterparts in Group 1.

Table 8 also shows that participants did not create any collections based on their music knowledge or preference in both groups where the system suggestions were available. Even though there is not sufficient evidence for why this is so, a speculation is that participants in those two groups considered MusicWiz's recommendations more valid or important than their personal interpretations of music. In addition, participants with insufficient knowledge about the music could easily associate the unknown songs with others by just following the similarity suggestions of the system. Another explanation is that creating a collection of disliked or out-of-interest songs is an easy way of avoiding the tedious process of classifying music that is hard to understand or to fit in an existing collection. It is possible that the Group 1 and 3 participants, without any support from the system, created collections like "Don't like", "Stuff I don't listen to" and "The ones I don't know" in an effort to save time from organizing songs hard for them to analyze and classify.

### 7.3.2 *Comments on System Features and Tasks*

Many of the responses to the open ended questions were comments made about specific features or activities. These provide insight into the difficulties encountered by the different groups.

Comments by participants in Group 1 mentioned the difficulty they had in assigning the same song to more than one collection (for task one) and to more than one playlist (for tasks two and three). The solution they had to use was to create copies of the provided song thumbnails. They also referred to the inconvenience of using a separate application for playing the music as well as the relatively long time (less than five seconds though) required by the MusicWiz player to upload a file before the playback is available (MusicWiz uploads the whole file into memory to allow time-based navigation). In task three, one Group 1 participant found the songs given as “seeds” for the creation of the three playlists too similar, making the selection of related music difficult.

Group 2 participants’ comments indicated how tedious it was to organize music that is not known and how helpful MusicWiz’s recommendations were in finding similar songs. They also expressed the desire to have direct access to all of the music as well as to be able to apply alternative organization schemes on the collection. The similarity metrics were also valuable for generating playlists. One participant commented that “I used the auto population (of the playlist) feature very successfully”. The same participant wished for more clarification of the playlist population menu as well as for the system to be able to keep the current population settings. The system

recommendations were also valuable for task three. Comments included “To create the playlist for the 4 Non Blondes – Misty Mountain Hop (one of the song-examples), I relied completely on the MusicWiz’s suggestions and they were actually quite good“, “Almost all of the songs that were used I got them by using the find similar songs. It was the easiest ...” and “I have used the find similar songs feature. Didn’t always agree with its findings but it always gave me a good start ...”. Participants in Group 3 complained about MusicWiz not providing the total length of the playlist as well as overlapping music previews that occurred due to abrupt switches of the mouse cursor from song to song.

Group 3 participants’ comments focused on their desire for more information about the songs and interaction issues in the workspace. They recommended the inclusion of additional data in the MusicWiz objects like information about the genre and the album. They mentioned the need for easier access to the object information, suggesting the ability to maximize and minimize objects similar to the capability provided for collections.

The feedback provided by Group 4 participants included suggestions for adding zoom in/out functionality to the MusicWiz workspace for easier overview and positive comments about playlist generation. Regarding task three, one participant expressed his satisfaction for the system support commenting that “The software made this very easy – I only made one change in each list so that they were just as I’d do them manually (or better)”.

Overall, comments regarding suggestions were very positive. Comments regarding the workspace indicated participants found the space useful and had suggestions for further capabilities.

## 8. CONCLUSION AND FUTURE WORK

Software supporting music management currently emphasizes the application and use of context-independent attributes of music files. While this metadata is valuable for locating specific files, it is not satisfactory for generating playlists automatically.

When encouraged to organize collections of music without using traditional metadata, participants used personal characteristics such as how well they knew or liked the song, memories they associated with the song, and their assessment of a song's mood or musical characteristics. Such assessments are highly personal and may even vary across time and contexts for the same user.

The results of the preliminary study showed that there are benefits and weaknesses in organizing personal music collections based on the context-independent metadata found in current tools and the malleable personalized interpretation found in spatial hypertext. Metadata provides for predictable access but the personal interpretation can capture characteristics that matter more when selecting music for playback.

Knowing ahead of time what characteristics of music are going to be important to a particular user is difficult. Most music-management systems support personal interpretation through the addition of new metadata fields and values. Users rarely add new attributes and values because of the effort required in the human-computer interface and because they may wish to express characteristics or concepts that are difficult to describe textually. Communicating that Dire Straits' *Sultans of Swing* is both upbeat and

mellow is not difficult with tags but indicating that it is not as upbeat as Cindy Lauper's *Girls Just Want to Have Fun* nor as mellow as Sarah McLachlan's *Angel* using traditional metadata turns the user into a knowledge engineer.

Such relative assessments of musical characteristics were part of why participants positively evaluated the ease of expression in spatial hypertext for personal interpretation. They found visual expression facilitated their interpretation of mood, memories, and musical dynamics. Yet, participants also indicated that the lack of views of their collection based on traditional metadata made it more difficult to locate songs that they knew they wanted. Visual personal interpretation, at least in the time-limited task of the study, enhanced users' expression but the resulting expressions were not always efficient representations for locating specific songs. The study also found that users view traditional metadata as insufficient for expressing their desires for playlists. Participants reported that creating playlists involves selecting music that includes variation yet fits well together. Six participants reported that they found visual expression in VKB useful as compared to their prior experiences organizing music collections.

These results indicate that the traditional metadata (artist, composer) is valuable for navigation of a collection but not for the direct specification of desired music and that the personal interpretation found in the visual expression was valuable for selection of music but not for navigation.

The MusicWiz personal music management environment was designed based on this feedback. It combines the easily expressed interpretations of music found in spatial

hypertext workspaces with the predictable and consistent explicit descriptions found in current metadata-based applications. In MusicWiz, users can associate songs by manipulating their representation in the workspace, can browse and retrieve music based on its lyrics, metadata values and melody features, and can navigate the collection according to the similarity of its content.

A comparative evaluation of MusicWiz's use of a spatial workspace for personal expression and its multi-faceted suggestions indicated positive results and areas for future work. Regardless of the type of the task they had to perform (classification, searching, or similarity assessment), the participants using the full system or features of it reported having better support, more fun and, in the case of the organization task, much faster completion times than the participants working on Windows Explorer. A wide range of criteria was used for the music classification during the organization task ranging from metadata values to music content, preference, knowledge and concept of listening. The participants with the system suggestions available were the only ones that did not create collections based on knowledge or preference trusting MusicWiz's recommendations in associating music. After-study comments confirm and enrich the results from the quantitative analysis valuing high the contribution of MusicWiz in the tasks. Participants found the system easy to use while they appreciated its ability to provide accurate and successful recommendations. Their concerns had to do with the desire for additional functionality (e.g. zooming in the workspace) and certain behavior issues of MusicWiz that do not undermine the value of including the workspace or



suggestions (e.g. playback of previews overlapping when the mouse was moved quickly across a number of songs.).

Aside from the techniques for music similarity and association evolved during the course of this research, three algorithms for creating multi-phrase song summaries were also developed and evaluated. The design of the algorithms reflects a range of approaches that vary between emphasizing the selection of repeated phrases and the selection of sonically different phrases. The study showed that participants believed that the multi-phrase summaries better represented the song than the introduction to the song. While the difference between the three algorithms was not significant, the results indicate a likely preference for algorithms that emphasize the selection of repeated phrases, at least in the genre of pop and rock where the structural components of the melody are more standardized and identifiable.

There are a number of features that can be developed in the future to increase MusicWiz's performance and applicability. To improve the accuracy of the provided recommendations, new modules of similarity assessment can be integrated utilizing information like usage statistics (recency and frequency of song access), user-provided ratings, input of song relatedness from internet radio and music recommendation websites like Pandora and Lastfm, and additional content-based features (e.g. sound or melody attributes). Towards the direction of providing community-based music recommendation services, MusicWiz could support the exchange of the extracted features and similarity assessments between users via a web-based application. Users in that group not only will be able to share their resources, knowledge and taste about

music but also to contribute in the creation of new, customized genres that will reflect the preference and perception of the music community they belong.

There is also future research to be done related to the music-summarization techniques. One of the complaints participants had during the task was that the switch between phrases in the summaries was too abrupt and hence distracting or even annoying. Use of phrase boundary detection for selecting the start of phrases could help as could the use of fade-in and fade-out effects. Based on participants' feedback about which parts / features of the songs are important, it would be interesting to examine if the integration of the introduction in the summaries can improve or accelerate the process of becoming familiar with new music. A comparison of the best of the proposed techniques with summaries containing only the most salient phrase of the song is also worth testing. Finally, it would be nice to improve the accuracy of the summarization approach to be applicable in genres like classical music and jazz where identification of the various themes and important components is more challenging.

Overall, this dissertation has explored the potential for improving the management and use of music collections. Techniques for supporting personal expression, multi-faceted suggestions and playlist generation, and music summarization have been developed and evaluated. These results provide a starting point for the design of both larger community-oriented music services that build on the current iTunes and Pandora style of interaction and more focused research into alternative techniques for the similarity assessment and summarization of music from different genres.

## REFERENCES

- Annesi, P., Basili, R., Gitto, R., and Moschitti, A. 2007. Audio Feature Engineering for Automatic Music Genre Classification. Proceedings of the RIAO. Pittsburgh, PA, [http://www.riao2010.org/old\\_riao-2007/papers/122.pdf](http://www.riao2010.org/old_riao-2007/papers/122.pdf) (Last accessed on 4/2010).
- Aucouturier, J-J., and Pachet, F. 2002. Music Similarity Measures: What's The Use? Proceedings of ISMIR. Paris, France, 157-163.
- Bartsch, M. and Wakefield, G. 2001. To Catch a Chorus: Using Chroma-Based Representation for Audio Thumbnailing. Proceedings of the Int. Workshop on Applications of Signal Processing to Audio and Acoustics. New Paltz, NY, 15-18.
- Bischoff, K., Firan, C., Nejdil, W., and Paiu, R. 2009. How Do You Feel about "Dancing Queen"?: Deriving Mood & Theme Annotations from User Tags. Proceedings of JCDL. Austin, TX, 285-294.
- Chai, W. and Vercoe, B. 2003. Music Thumbnailing via Structural Analysis. Proceedings of ACM Multimedia. Berkeley, CA, 223-226.
- Chen, Y., and Butz A. 2009. Musicsim: Integrating Audio Analysis and User Feedback in an Interactive Music Browsing UI. Proceedings of IUI. Sanibel, FL, 429-434.
- Cheveigne, D. A., and Kawahara, H. Y. 2002. YIN, A fundamental frequency estimator for speech and music. Journal of the Acoustical Society of America. 111(4), 1917-1930.
- Cooper, M., and Foote, J. 2002. Automatic Music Summarization via Similarity Analysis. Proceedings of the 3<sup>rd</sup> Intl. Symposium on Music Information Retrieval. IRCAM – Centre Pompidou, Paris, France, 81-85.
- Crossen, A., Budzik, J., and Hammond, J. K. 2002. Flytrap: Intelligent Group Music Recommendation. Proceedings of IUI. San Francisco, CA, 184-185.
- Cunningham, S. J., Jones, M., and Jones, S. 2004. Organizing Digital Music for Use: An Examination of Personal Music Collections. Proceedings of ISMIR. Barcelona, Spain, 447-454.
- Downie, S., and Hu, X. 2006. Review Mining for Music Digital Libraries: Phase II. Proceedings of JCDL. Chapel Hill, NC, 196-197.

- Downie, S., West, K., and Hu, X. 2008. Dynamic Classification Explorer for Music Digital Libraries. Proceedings of JCDL. Pittsburg, PA, 422-422.
- Francisco-Revilla, L. and Shipman, F. 2004. Instructional Information in Adaptive Spatial Hypertext. Proceedings of ACM Document Engineering, Milwaukee, WI, 124-133.
- Foote, T. J. 1997. Content-based Retrieval of Music and Audio. Proceedings of SPIE. San Diego, CA, 138-147.
- Hanna, P., Robine, M., and Rocher T. 2009. An Alignment Based System for Chord Sequence Retrieval. Proceedings of JCDL. Austin, TX, 101-104.
- Hoashi, K., Zeitler, E., and Inoue, N. 2002. Implementation of Relevance Feedback for Content-based Music Retrieval Based on User Preferences. Proceedings of SIGIR Tampere, Finland, 385-386.
- Goto, M. and Goto, T. 2005. Musicream: New Music Playback Interface for Streaming, Sticking, Sorting, and Recalling Musical Pieces. Proceedings of ISMIR. London, UK, 404-411.
- Kim, S., Kim, S., Kwon, S., and Kim, H. 2006. A Music Summarization Scheme Using Tempo Tracking and Two Stage Clustering. Proceedings of the IEEE 8<sup>th</sup> Workshop on MMSP. Victoria, BC, Canada, 225-228.
- Knees, P., Schedl, M., Pohle, T., and Widmer, G. 2007. Exploring Music Collections in Virtual Landscapes. Proceedings of Multimedia. Augsburg, Germany, 14(3), 46-54.
- Knuth D. 1973. The Art of Computer Programming - Volume 3: Sorting and Searching. Addison-Wesley Publishing Company, Reading, MA.
- Kuo, F., and Shan, M. 2004. Looking for New, Not Known Music Only: Music Retrieval by Melody Style. Proceedings of JCDL. Tucson, AZ, 243-251.
- Leitich, S., and Topf, M. 2007. Globe of Music - Music Library Visualization Using GeoSOM. Proceedings of ISMIR. Vienna, Austria, 167-170.
- Li, Q., Kim, B M., Guan, D H, and Oh D W. 2004. A Music Recommender Based on Audio Features. Proceedings of SIGIR. Sheffield, UK, 532-533.
- Lillie, A. 2008. MusicBox: Navigating the space of your music. Thesis, Massachusetts Institute of Technology, School of Architecture and Planning, Cambridge, MA, <http://hdl.handle.net/1721.1/46583> (Last accessed on 5/2010).

- Liu, D., Lu, L., and Zhang, H. 2003. Automatic Mood Detection from Acoustic Music Data. Proceedings of ISMIR. Baltimore, MD, 81-87.
- Logan, B., and Chu, S. 2000. Music Summarization Using Key Phrases. Proceedings of IEEE ICASSP. Istanbul, Turkey, 749-752.
- Logan, B., Kositsky A., and Moreno P. 2004. Semantic Analysis of Song Lyrics. Proceedings of IEEE ICME. Taipei, Taiwan, 827-830.
- Logan, A., and Salomon, A. 2001. A Music Similarity Function Based on Signal Analysis. Proceedings of IEEE ICME. Tokyo, Japan, 745-748.
- Lu, L., and Zhang, H. 2003. Automated Extraction of Music Snippets. Proceedings of the ACM Multimedia. Berkeley, CA, 140-147.
- Mardirossian, A., and Chew, K. 2006. Music Summarization via Key Distributions: Analyses of Similarity Assessment Across Variations. Proceedings of ISMIR. Victoria, British Columbia, Canada, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.1579&rep=rep1&type=pdf> (Last accessed on 4/2010).
- Marshall, C.C., and Shipman, F. 1995. Spatial Hypertext: Designing for Change. Communications of the ACM, 38, 8, ACM Press. New York, 88-97.
- Monge, E. A., and Elkan, C. 1996. The Field Matching Problem: Algorithms and Applications. Proceedings of the Second Conference on Knowledge Discovery and Data Mining. Portland, OR, 267-270.
- Neumayer, R., Dittenbach, M., and Rauber, A. 2005. PlaySOM and PocketSOMPlayer: Alternative Interfaces to Large Music Collections, Proceedings of ISMIR, London, UK, 618-623.
- Ong, B. 2006. Structural Analysis and Segmentation of Music Signals. Dissertation submitted to the Department of Technology of Universitat Pompeu Fabra.
- Pampalk, E., Rauber, A., and Merkl, D. 2002. Content-based Organization and Visualization of Music Archives. Proceedings of ACM Multimedia. Juan-les-Pins, France, 570-579.
- Pardo, B., Little, D., Jiang, R., Livni, H., and Han, J. 2008. The Vocalsearch Music Search Engine. Proceedings of JCDL. Pittsburgh, PA, 430-430.
- Peeters, G., Burthe, A., and Rodet, X. 2002. Toward Automatic Music Audio Summary Generation from Signal Analysis. Proceedings of the ISMIR. Paris, France, 94-100.

- Rabiner, L. and Juang, B. 1993. *Fundamentals of Speech Recognition*. Prentice-Hall. Englewood Cliffs, NJ.
- Salton, G., Wong, A., and Yang, C.S. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, vol. 18, iss. 11, 613-620.
- Shao, X., Maddage, N., Xu, C. and Kankanhalli, M. 2005. Automatic Music Summarization Based on Music Structure Analysis. *Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*. Philadelphia, PA, 169-172.
- Shipman, F. M., Marshall, C. C., and Moran, T. P. 1995. Finding and Using Implicit Structure in Human Organized Spatial Layouts of Information. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Denver, CO, 346-353.
- Shipman, F. M., Hsieh, H., Airhart, R., Maloor, P., Moore J.M., and Shah, D. 2001b. Emergent Structure in Analytic Workspaces: Design and Use of the Visual Knowledge Builder. *Proceedings of IFIP INTERACT'01: Human-Computer Interaction*. Tokyo, Japan, 132-139.
- Shipman, F., Hsieh, H., Airhart, R., Maloor, P., and Moore, M. J. 2001c. The Visual Knowledge Builder: A Second Generation Spatial Hypertext. *Proceedings of Hypertext*. Aarhus, Denmark, 113-122.
- Shipman, F., and Marshall, C.C. 1999. Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems. *Journal of the Computer Supported Cooperative Work*, vol. 8, 333-352.
- Tsai, W., and Wang, H. 2005. On the Extraction of Vocal-related Information to Facilitate the Management of Popular Music Collections. *Proceedings of JCDL*. Denver, CO, 197-206.
- van Breemen, A., and Bartneck, C. 2003. An Emotional Interface for a Music Gathering Application. *Proceedings of IUI*. Miami, FL, 307-309.
- van Gulik, R., Vignoli, F., and van de Wetering, H. 2004. Mapping Music In the Palm of Your Hand, Explore and Discover Your Collection. *Proceedings of ISMIR*. Barcelona, Spain, 409-414.
- Xu, C., Maddage, N., and Shao, X. 2005. Automatic Music Classification and Summarization. *IEEE Transactions on Speech & Audio Processing*. 13 (3), 441-450.
- Zadel, M., and Fujinaga, I. 2004. Web Services for Music Information Retrieval. *Proceedings of ISMIR*. Barcelona, Spain, 478-483.

## VITA

Name: Konstantinos A. Meintanis

Address: Department of Computer Science & Engineering,  
Texas A&M University, TAMU 3112, College Station, TX 77843

Email Address: meinkos@yahoo.com

Education: B.S., Informatics, Athens University of Economics & Business,  
Greece, 2001  
Piano Performance Degree, Hellenic Conservatory, Greece, 2002  
Ph.D., Computer Science, Texas A&M University, 2010