

The Semantic Mp3 Player

A semantic music media prototype

Fredrik Johansson & Petter Hagman
dit02fjs, dit02phn [@cs.umu.se](mailto:dit02fjs@cs.umu.se)

Department of Computing Science
Umeå University
2007-07-11

Introduction.....	3
Implementation.....	3
Vision.....	3
Semantic descriptors bottom-up computation module	4
Music similarity.....	4
Music collection structuring and navigation module	5
GUI (Graphic User Interface).....	5
Interface interaction.....	7
Discussion.....	8
References	8

Introduction

Our goal with this project is to construct a prototype for a semantic mp3-player. We want this device to be smart enough to classify and categorize different audio tracks into different genres and subgenres. Apart from these features it should also be capable of understanding what the user wants to listen to. The prototype itself will not be able to do any of these things, but it will give a complete visualization of how these different tasks and features would work. How this would actually work will be more discussed in this rapport.

Implementation

The mp3-player prototype will be created in Flash CS3 and it will contain a partial solution to the whole mp3-player. During the test phase of the construction it would have been great to actually test the interface on a real handheld device in order to see how the visual elements worked in different kinds of contexts. Though the Flash CS3 program has a built in development center for handheld devices it doesn't support testing and debugging programs that will run on handheld devices such as Pocket PCs with the Windows Mobile 5.0 operative system. This is unfortunate since the Semantic Mp3 player have to run on such devices. The reason for this is that the device has to have a touch screen so that the user can interact with the virtual music elements.

Vision

Our vision is a player that actually can find a semantic resemblance between tracks stored on a specific device. The question is how this will work. During the design phase the group searched the web for similar projects and even crossed into domains that had some small elements that could be an influence that could be used in the later phases. Some of the ideas on how to create a semantical content on a device containing encoded music data have been inspired by a project similar to ours. The project is called SIMAC and it stands for Semantic Interaction with Music Audio Content [1]. The SIMAC project was funded by the European Commission and delivered their final reports in 2006. Since this date many manufacturers have been eager to use the components created during this project to enhance the experience user have when listening to music.

Below some different approaches on how to create semantic metadata from music content stored on a handheld device. The different methods determine what kind of information that can be extracted from the audio files. How the audio files will be read to a computer is the interesting in all of this. There is actually a great deal of information that can be extracted from an audio file, some information is more important than other. I.e. Typical information that can characterize a category of music is bpm (Beats Per Minute) and distinction of what kind of instruments that are

involved. When trying to explain all of this information to a computer a bottom up semantic descriptor can be used.

Semantic descriptors bottom-up computation module

Bottom up semantic descriptors of music are measures that can be calculated from audio signals using signal processing, artificial intelligence and musical knowledge. These descriptors can find the musical attributes of audio signals (such as chords, rhythm, instrumentation etc) giving it a higher level of semantics (compared to low-level features like spectral coefficients).

- Rhythm-related descriptors identify temporal periodicities, patterns and clusters of temporal events, which characterise the rhythm of a musical piece in a digital recording.
- Harmony-related descriptors characterise the harmony and tonality of a musical piece (also related to the melodic content).
- Instrumentation/timbre descriptors are intended to characterise the quality (and possibly the quantity) of the instrumentation in a musical recording. The timbre can be explained as a combination of attributes of a sound that distinguishes it from other sounds of the same pitch and volume.
- Acoustic descriptors refer to certain characteristics of the sound that are related to the acoustic properties of a musical recording and are can be discovered from determining the intensity and the spatial attributes of the signal.
- Segmentation descriptors identify the long-term structure of a musical piece (e.g. verse-chorus-verse-bridge-chorus) by finding blocks of common attributes using mostly low-level feature representations of music signals. This can be used as prior information for other identification tasks, such as instrumentation and dynamics, and as corroborative evidence for rhythm and tempo analysis.
- Complexity descriptors are associated to one or more musical dimensions, and characterise the effort or the amount of knowledge to be used by the listener to understand the information in the signal. They can also be seen as quantifying the nature (and size) of the “palette” used to compose the music within a song.
- Music Similarity within categories refers to those attributes that groups of songs (e.g. belonging to the same genre) have in common. Due to its nature, music similarity only has validity at collection level.

Music similarity

There are many aspects of similarity (timbre, harmony, rhythm, etc.), and there are different sources from which these can be computed (audio, web-pages, lyrics, etc.). Most of all, similarity is a perception which depends on the listener point of view and context. The main scope is to define similarity as the concept which pieces within a

genre (or subgenre) have in common. This allows highly efficient (i.e. fast and cheap) evaluations of the similarity measures (since genre labels for artists are readily available).

We primarily use nearest neighbour classifiers (and genre classification) to evaluate the similarity measures. The idea is that pieces within the same genre should be very close to each other. In addition we use inter and intra group distances. These are particularly useful in understanding how well each group is modelled by the similarity measure. To evaluate similarity measures the first half of each song is compared to the second half. The idea is that a good similarity measure would recognize these to be highly similar.

Music collection structuring and navigation module

Assume that there are "lazy" users who do not wish to select each piece they want to listen to individually. This user wants to interact with the music device as little as possible, but might be equally as determined what kind of music he/she wants to listen to. This user needs an automatically generated playlist based on a few simple inputs from the user. These playlists can be dynamically updated (and improved) based on minimum user feedback, and they can be initialized as simply as turning the tuner on a radio. These interfaces support the exploration and navigation of music collections while at the same time enjoying listening to music.

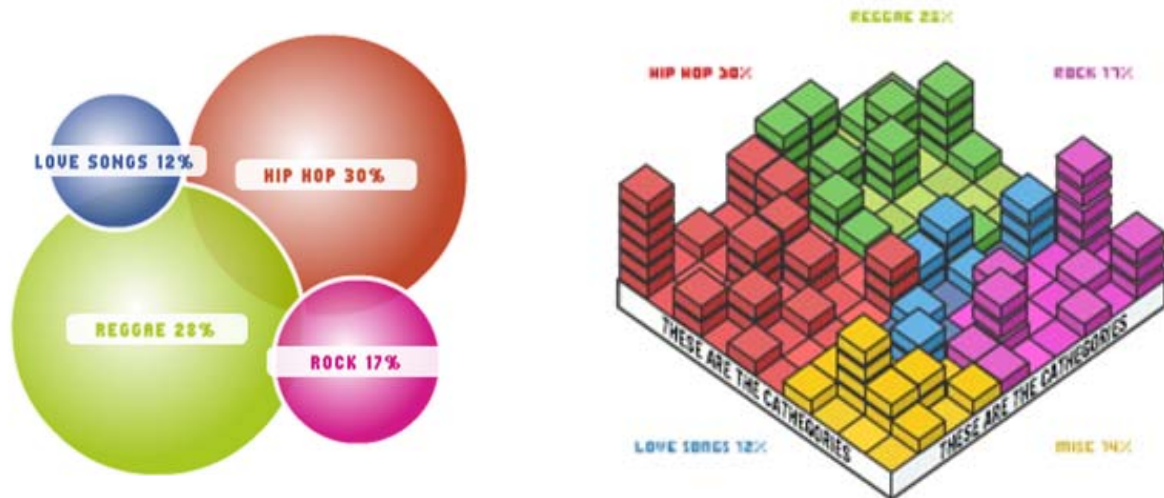
Common approaches to creating playlists are to randomly shuffle a collection (e.g. iPod shuffle) or manually select songs. The prototype developed in this project will instead use heuristics to adapt playlists automatically given a song/category to start with (seed song/category) and immediately give the user feedback that the user can evaluate. The users on the other hand give feedback back to the system by pressing a skip button if they dislike the current song. Songs similar to skipped songs are removed, while songs similar to accepted ones are added to the playlist. The heuristics embedded in the prototype will be evaluated with hypothetical use cases. For each use case, specific user behaviour (e.g. the user always skips songs by a particular artist) is assumed. Using this technique with audio similarity and simple heuristics the number of necessary skips can be reduced drastically.

GUI (Graphic User Interface)

During this project a prototype has been created. This prototype mainly serves the purpose of illustrating how the interface will handle the abstraction from commonly used playlists to visually categorized and structured content. The prototype was created in Adobe Flash CS3 and targets handheld devices with touch screen support.

During the process of creating the prototype many different approaches were discussed and investigated. After a while a number of key elements began to emerge and the main concepts for the prototype were established. This main idea for the

prototype was to illustrate the content available on the device in a simplistic and easily viewed manner. The images shown below are two different approaches to viewing the content.



Even though the left one was friendlier to the eye, we wanted the illustration to show precise information that the user could also interact with. The circles couldn't support this kind of interaction because the borders are circular and the fact that content won't form perfect circular forms as the ones depicted. The concept chosen was instead the one on the right. This concept supports user interaction and also delivers a somewhat full and correct content visualization.

The user content visualization displays major genres and subgenres instead of files and directories. Each block represents one percent of the content stored on the device the prototype is running on. A staple of blocks represents a typical subgenre and the colour code represents a major genre. When a user clicks a subgenre a list of artists typical for this subgenre is displayed. Here the user can choose to add this subgenre to the playlist. Compared to other semantic music projects this prototype uses a seed-genre instead of a seed-song. This lifts the entire experience to a higher hierarchical level and can therefore cover a larger number of songs so that a single user interaction can create a playlist sufficiently large with good semantic similarity. The user can also choose to enlarge the playlist to cover sub genres which are close in semantic proximity.

The visual representation of the content is a 10x10 squared board that equals 100 percent of the music collection. The content boards can display four major genres and one miscellaneous genre that contain all genres that are too small to be displayed as their own major genre. How the major genres are displayed depends on their similarity to other major genres and the same goes for sub genres. Sub genres with high similarity are placed close together inside the major genre they derive from. These inter and intra group distances are calculated correctly as far as possible, but

due to the limited space the distances can't be considered correct in all cases, but can be considered as a visual clue to the user.



Sequential screenshots of creating a playlist

Interface interaction

When a user has created a playlist and the music plays the prototype offers a number of different interactions. The song currently playing is situated in the center and the blocks on left and right side is songs coming up next and songs previously played. The user can click these boxes to jump to any of these songs. Each box is coloured in the colour of the category they resides within. The three buttons below is:

- **The Love button**
 - Adds the current track to a "Users Favourites" list and the program simultaneously searches for content similar to this song.
- **The Ban button**
 - This buttons removes the current track from the playlist and skips the song. Similar songs are also removed from the playlist.
- **The Skip button**
 - This buttons skips to the next song



Discussion

Now that we have presented these technologies and methods we can discuss whether it is a good invention for mobile devices or not. Compared to the SIMAC project (that is meant to be on a computer with internet access) that can “foaf” the music for you and recommend new artists by accessing music databases situated on the net, it might not be equally useful. But still, some of their techniques regarding semantics behind the songs and genres, the classification technique and the playlist generator are all good ideas that are very much implementable and very interesting for the mobile market.

Since major parts of the music stored on handheld devices is compressed into the mp3 format advantages can be taken from the ID3v2 tags already implemented in these files. A lot of semantic data useful to the prototype can be stored in these tags.

Since mp3 players often lack the computer processability that stationary computers have it could be a good idea to pre-process the audio files before storing them on the handheld device. When doing this the handheld device doesn't have processing so much data and therefore doesn't consume as much battery power. All this goes in line with the fact that handheld devices is becoming smaller and smaller.

References

- [1] SIMAC (Semantic Interaction with Music Audio Content)
<http://www.semanticaudio.org>
[Accessed 8/5 2007]