THE CUIDADO MUSIC BROWSER: AN END-TO-END ELECTRONIC MUSIC DISTRIBUTION SYSTEM

François Pachet, Jean-Julien Aucouturier, Amaury La Burthe, Aymeric Zils, Anthony Beurive

SONY Computer Science Laboratory
6, rue Amyot 75005 Paris, France
+33 144 080 516

{pachet,jj,amaury,aymeric,beurive} @csl.sony.fr

ABSTRACT

The IST project Cuidado, which ran from January 2001 to December 2003, produced the first entirely automatic chain for extracting and exploiting musical metadata for browsing music. The Sony CSL laboratory is primarily interested in the context of popular music browsing in large-scale catalogues. First, we are interested in human-centred issues related to browsing "Popular Music". Popular here means that the music accessed to is widely distributed, and known to many listeners. Second, we consider "popular browsing" of music, i.e. making music accessible to non specialists (music lovers), and allowing sharing of musical tastes and information within communities, departing from the usual, single user view of digital libraries. This research project covers all areas of the music-to-listener chain, from music description - descriptor extraction from the music signal, or data mining techniques -, similarity based access and novel music retrieval methods such as automatic sequence generation, and user interface issues. This paper describes the scientific and technical issues at stake, and the results obtained.

1. INTRODUCTION

1.1. Existing Popular Music Access Systems

There are now many on line searchable music databases. We can classify them in the following categories.

First, purely editorial systems propose systematic editorial information on popular music, including albums track listings (CDDB[1], Musicbrainz[2]), information on artists and songs (AMG[3] and Muze[4]). This information is created by music experts, or in a collaborative fashion (CDDB, Musicbrainz). These systems provide useful services for *Electronic Music Distribution* (EMD) systems, but cannot be considered as fully-fledged EMD systems *per se*, as they provide only superficial and incomplete information on music titles, supposed to exist somewhere else.

The MoodLogic[5] browser proposes a complete solution for Popular Music access. The core idea of MoodLogic is to associate metadata to songs automatically thanks to two basic techniques: 1) an audio fingerprinting technology able to recognize music titles on personal hard disks, and 2) a database collecting user ratings on songs, which is incremented automatically, and in a collaborative fashion. An ingenious proactive strategy is enforced to encourage users to rate songs, in order to get tokens that allow them to get more metadata from the server. MoodLogic relies entirely on metadata obtained from user ratings and does not perform any acoustic analysis of songs. However, collaborative music rating does not exhaust the description potential of music, and our Browser proposes many other types of metadata.

Other proposals have been made either for fully-fledged music browsers, or for ingredients to be used in browsers (fingerprinting techniques, collaborative filtering systems, metadata repositories, e.g. Wold et al. [20]) that we cannot cover here for reasons of space. We will describe in this paper only the parts of our project that we think are original and may contribute to address the needs of our targeted users.

1.2. The Cuidado Music Browser

The Cuidado music browser aims at developing all the ingredients of the music-to-listener chain, for a fully-fledge content-based access system. More precisely, the project covers the areas of 1) editorial metadata, 2) acoustic metadata, 3) metadata exploitation and browsing tools, 4) management and share of metadata among users

The next sections describe the most important results obtained for each of these aspects.

2. EDITORIAL METADATA

To manage collections of music titles an application must have access to many information to identify,

1. http://www.gracenote.com/
2. http://www.musicbrainz.org/
3. http://www.allmusic.com/
4. http://www.muze.com/
5. http://www.moodlogic.com/

categorize, index, classify and generally organize music titles.

We consider here two types of data as editorial metadata:

- Consensual information or facts about music titles and artists,
- Content description of titles, albums or artists.

The first category is common to already existing EMD systems and does not raise any particular problem, as this information is universal by nature. It includes for instance: artist and songs name, albums and tracks listing, group members , date of recording for a given title, short biography for artists with date of birth, years of activity, etc.

The second category is more problematic. Content description includes such widely needed information as artist style, artist instruments, song mood, song review, song or artist genre and more generally attributes aiming at describing the intrinsic nature of the musical item at stake (artist or song). These descriptions are useful to the extent that they can be used for musical queries in large catalogues. The music browser enables to issue queries for both categories.

Furthermore, the music browser has a tool (see figure 1) devoted to editorial information management. The global architecture of the system is detailed in section 6. This tool allows editing and adding artists and/or songs properties.

2.1. Editorial metadata philosophy

Editorial metadata are associated distinctly with music titles and artists.

Artists (taken in the most general sense) are key *music identifiers* for many users: Yesterday is by "The Beatles", and "The 5th symphony" is by Beethoven. Artists are used also for solving ambiguity: "With a Little Help from my Friends" by the Beatles, is definitely not the same tune as the version by Bruce Springsteen. The "Stabat Matter" by Pergolese is not the one by Boccherini, etc. We call these artists "primary artists" as they are most commonly used to identify music titles. These examples show that primary artists are common ways of identifying music titles but also that the role of primary artists changes with styles: in Classical music, primary artists are usually composers. In non Classical music they are usually performers. In our Browser, we introduced the notion of primary artists in a deliberate ambiguous way, to cope for Classical and non Classical music in a uniform way.

There are cases where primary artists are not enough for characterizing the identity of a piece. The "1st partita" of Bach has been recorded by Glenn Gould, and also by many other pianists, and this distinction is of course very important: not only for interpreters, but also for conductors (for orchestral pieces). In non-Classical music the need for secondary artists is also obvious, for instance to indicate that the Springsteen version of "A little help" is indeed a Beatles song.

Existing repositories of editorial information do not provide systematic schemes for accessing artists and their relations to songs. This led us to constitute a database of artists, or more generally of "Musical Human Entities" (MHE), including both performers, composers, but also groups (the Beatles), orchestra (the Berlin Philharmonic), duets (Paul McCartney & Michael Jackson). To each artist (or MHE) is associated a limited but useful set of properties in fixed ontologies: type (composer, singer, instrumentist, etc.), country of origin, language (for singers), type of voice (for singers also), main instrument (for instrumentists). Other information concern the relation MHE entertain with each other. For instance, Paul McCartney is a *member of* The Beatles, and artist Phil Collins a *member of* the group Genesis. The Editorial MHE database may be seen more as a knowledge base than a database.

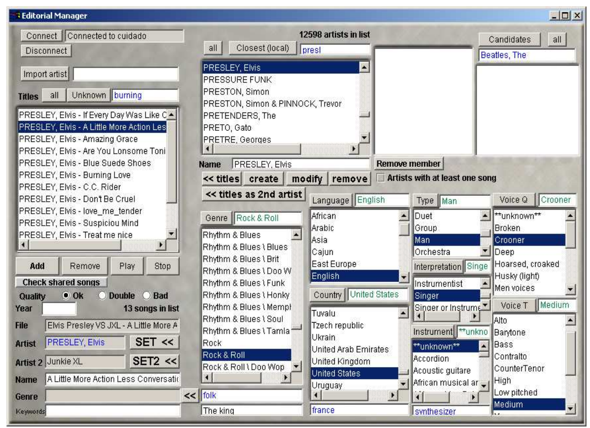

Figure 1 – The editorial data management panel

Concerning music titles, our tool enables basic editions as title name or keywords, as well as less obvious features such as title genre, primary and secondary artist introduced before.

Both artists and songs can be associated with a specific genre. Genres are badly needed for accessing music, and are as badly ill-defined. Our studies on existing taxonomies of genres have shown that there is no consensus, and that a consensus is probably impossible [4]. However, we propose here several ways to partially solve this problem. After several years of trials [15] and errors, we ended up with a simple two-level genre taxonomy consisting of 250 genres. The main property of this taxonomy is flexibility: users can classify artists or songs either in a generic way (Classical, Jazz), more precisely (Jazz / BeBop, Classical/Baroque). However, simpler taxonomies may also produce frustration, as some categories may contain artists or songs that users would consider very different. To make our taxonomy more flexible, we have introduced an optional "keyword" field, which may contain free words. These words may be entered by users to further refine their own classification perspective on artists or songs. This simple yet flexible approach has the advantage of uniformity: artists and songs are classified in the same taxonomy, allowing for various degrees of precision. For instance, The Beatles is classified in “Pop / Brit”, but Beatles songs may be classified in other genres (e.g. “Revolution 9” is “Rock / Experimental”).

Figure 2 – the “member_of” predicate

3. ACOUSTIC METADATA

The main type of metadata that the MB proposes for songs besides editorial information is acoustic metadata, i.e. information extracted from the audio signal. The Mpeg7 standard aims at providing a format for representing these information, and a specialized audio group produces specific constructs to represent musical metadata [1,10]. However, music metadata in Mpeg7 refers in general to low-level, objective information that can be extracted automatically in a systematic way. Typical descriptors (called LLD for Low-Level Descriptors in the Mpeg7 jargon) proposed by Mpeg7 concern superficial signal characteristics such as means and variance of amplitude, spectral frequencies, spectral centroid, ZCR (zero crossing rate), etc.
Concerning high-level descriptors that can be mapped to high-level perceptual categories, Mpeg7 is strictly concerned with the format for representing this information, and not the extraction process *per se*.

3.1. Extracting High-Level Music Percepts

We have conducted in the project several studies focusing on particular dimensions of music that are relevant in our context.

3.1.1. Rhythm

We have proposed a rhythm extractor [22], that is able to extract the time series of percussive sounds in music signals of popular music. Rhythm information is a useful extension of tempo or beat, as proposed by Scheirer in [17]. However, many things remain to be done in the field of rhythm. One key issue seems to rely not so much in how to extract rhythm, but how to exploit the information: most people are unable to describe rhythm with words, and even less to produce rhythm (our attempts at designing a query by rhythm did not prove successful).

3.1.2. Energy

In [21], we have addressed another dimension of music pertaining to popular music access, the perceptual energy, i.e. whether a song is thrilling and exciting (e.g. hard rock, dance music), or relaxing and calm (e.g. a piano piece by Schumann).
We have studied the correlation of experimental measures (user tests) with a variety of signal features, such as tempo, raw signal energy, spectral analysis, the associated variances, correlations... as well as their linear combinations (using discrimination analysis) and their possible compositions with signal operators (filters, etc…). The most discriminative parameter we found is $\log 10(\mathrm{var}(\mathit{diff}\,(x^2)))$, which gave a classification error of 22% on the validation set.

3.1.3. Timbre

In [2], we have proposed to describe music titles based on their global *timbral quality*. Our motivation is that, although it is difficult to define precisely music taste, it is quite obvious that music taste is often correlated with timbre. Some sounds are pleasing to listeners, other are not. Some timbres are specific to music periods (e.g. the sound of Chick Corea playing on an electric piano), others to musical configurations (e.g. the sound of a symphonic orchestra). In any case, listeners are sensitive to timbre, at least in a global manner.
We model the global “sound” of a music title as a distribution in the space of mel cepstrum coefficients (MFCC). MFCCs provide a compact representation of the signal’s spectral envelopes, which are a good correlate of the timbre. By comparing timbre distributions between titles, it is then possible to match music titles of possible very different genres based solely on their timbre color. Figure 3 shows a 3D projection of the feature space (which is originally of dimension 8), showing two distributions of MFCCs, each modelled with a mixture of 3 gaussian distributions (GMM). The light-grey GMM is the timbre model of the song “The Beatles – Yesterday”, and the dark-grey GMM is the timbre model of the song “Joao Gilberto – Besame Mucho”. This two songs have a very similar “sound” (acoustic guitar and a string quartet, plus a gentle and melancholic male voice), and indeed we see that their MFCC distributions are very close. As explained in section 4, timbre models are used in the MusicBrowser to compute similarities between songs.

3.1.4. Instrumental/Voice presence

A fourth descriptor which is currently available in the Music Browser describes whether a given tune contains singing voice or only instrumental sounds. This property is useful e.g. to either access particular “genres” of music (“opera” falls in the first category, while “piano sonatas” falls in the instrumental category), or to differentiate different versions of the same song (e.g. “Dub” instrumental versions of “reggae” songs).

There has been a large number of studies about the issue of speech/music discrimination (see e.g. [18]), which has received successful solutions, but the detection of singing voice has proved a more difficult problem. Berenzweig in [7] proposes to use complex features (output probabilities of a speech recognizer system) combined with hidden Markov models (HMMs). The extractor currently used in the Music Browser was designed automatically by the EDS system, described in the next section. It has a classification error of 19% on the validation set.

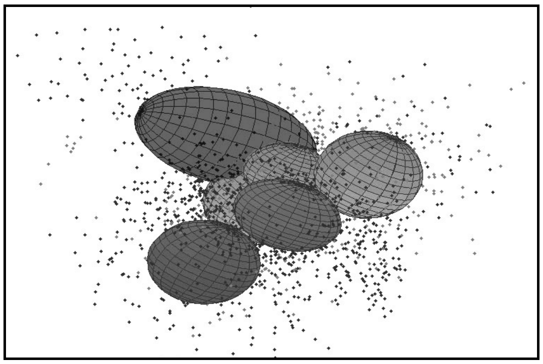

Figure 3: comparison of the timbre models of two songs: "The Beatles - Yesterday" and "Joao Gilberto – Besame Mucho"

3.2. EDS: A General Framework for Extracting Extractors

These various studies in descriptor extraction from acoustic signals have shown that the design of an efficient acoustic extractor is a very heuristic process, which requires sophisticated knowledge of signal processing, intuitions, and experience. Indeed, most approaches in feature extraction as published in the literature consist in using statistical analysis tools to explore spaces of combinations of LLD. The approaches proposed by Peeters [16], Scheirer [18] and Tzanetakis [19] typically fall in this category. However, these approaches are not capable of yielding precise extractors, and depend on the nature of the palette of LLD, which usually do not capture the relevant, often intricate and hidden characteristics of audio signals. Consequently, designing extractors is very expensive and hazardous.

On the other hand, user studies have shown that there is a virtually infinite number of extractors of musical attributes that could be useful in EMD systems. Different users have different needs: one – say, a jazz musician - might be interested in listening to songs which exhibit a particular chord sequence, another may be interested by the sound ("some saturated guitar with a little bit of chorus"), while another simply wants to find "funky" music for his birthday party. Even when talking about the same attribute, the definitions (i.e. in terms of pattern recognition, the training sets) vary a lot. The perception of "harmonic complexity" of a tune for instance highly depends on the musical expertise of the listener.

These experiments have given rise to a systematic approach to feature extraction, embodied in the EDS system [12]. Departing from the usual LLD approach, the idea of EDS is to automate – in part or totally – the process of designing extractors. EDS searches in a richer and more complex space of signal processing functions, much in the same way than experts do: by inventing functions, computing them on test databases, and modifying them until good results are obtained.

To reach this goal EDS uses a genetic programming engine, augmented with fine grained typing system, which allows to characterize precisely the inputs and outputs of functions. EDS also uses rewriting rules to simplify complex signal processing functions (see the example of the Perceval equality being used by EDS to simplify the expression in Figure 4). Finally EDS uses expert knowledge to guide its search, in the form of heuristics.

Typical heuristics include "do not try functions which contain too many repetition of the same operator", or "apply twice a FFT on a signal is interesting, but not 3 times", or also "spectral coefficients are particularly useful when applied on signals in the temporal domain, possible filtered", etc.

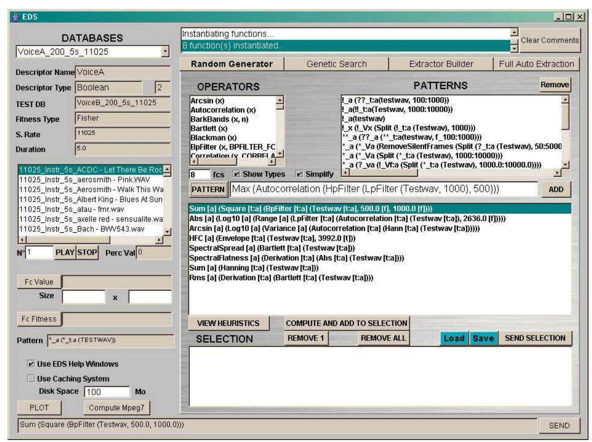

Figure 4: Screenshot of the EDS system.

The signal operators available in the EDS which serve as basic bricks for building extractors include the full set of MPEG7 LLDs, but also typical signal operators like filters, FFTt, time windowing, and higher level operators like pitch detection, partial tracking or mel filterbank. These operators are selected from the literature and our experiments of designing extractors manually. The features designed and discovered by the system can be further combined, manually or automatically, by statistical models like GMMs or HMMs, or classifiers like neural networks. The output of the whole process is an executable file, which can be directly integrated in applications like the Music Browser.

The current extractors targeted by EDS are perceptual energy (or a refinement of the descriptor we designed by hand), discrimination between songs and instrumental (already described in the previous section), discrimination between studio and live versions of songs, harmonicity vs noisiness, percussivity, harmonic complexity, etc. The ambitious goal of EDS makes it a project in itself, as it aims at capturing complex knowledge, in an expanding field. However, we think that the contribution to the MIR community is potentially important as it is a first step towards a unified vision of high level audio feature extraction.

4. SIMILARITY

The notion of similarity is of utter importance in the field of music information retrieval, and the expectation to have systems that find songs that are "similar" to one or several seed songs is now second nature. However, here again, similarity is ill-defined, and it can be of many different sorts. For instance, one may consider all the titles by a given artist as similar. And they are, of course, artist-wise. Similarity can also occur at the feature level. For instance, one may consider that Jazz saxophone titles are all similar. Music similarity can yet occur at a larger level, and concern songs in their entirety. For instance, one may consider Beatles titles as similar to titles from, say, the Beach Boys, because they were recorded in the same period, or are considered as the same "style". Or two titles may be considered similar by a user or a community of users for no objective reason, simply because they think so.

4.1. Acoustic similarity

Feature-based similarity is trivially obtained by defining similarity measures from the metadata obtained and described above, either editorial or acoustic. Most descriptors yield implicit similarity measures that can be useful in some circumstances, e.g. similarity of tempo, of energy, or similarity based on artist relationships, etc.

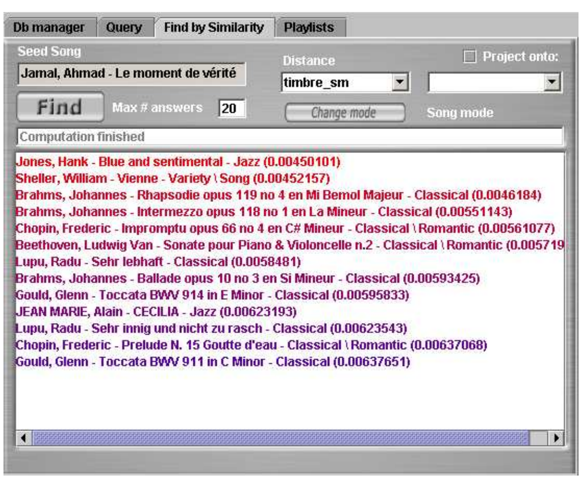

Figure 5: the "Find by Similarity" panel in the Music Browser

One very interesting type of similarity that we already mentioned is based on the global "timbre" of the songs. The distance analysis is based on Gaussian models of Cepstrum coefficients as described in [2]: a first model is sampled and then the likelihood of the samples is computed given the other model. Figure 5 shows a screenshot of the "Find by Similarity" panel in the Music Browser. Here, the user has select a jazz piano song ("Ahmad Jamal- L'instant de Vérité"), and asked the system to return "songs that sound the same". The result lists contains songs of many genres, which all contain romantic-styled piano: Jazz (Hank Jones, Alain Jean-Marie), Classical piano pieces (Brahms, Chopin), and even a "Variety" song (William Sheller, a French singer and pianist who had a classical training).

4.2. Cultural Similarity

Cultural similarity is based on a well-known technique used in statistical linguistics: co-occurrence analysis. Co-occurrence analysis is based on a simple idea: if two items appear in the same context, it is obvious that there is some kind of similarity between them. In linguistics, co-occurrence analysis based on large corpora of written and spoken text has been used to extract clusters of semantically related words. Similarity measurements based on co-occurrence counts have been demonstrated to be cognitively plausible [8]. We have identified several interesting corpora:

- The web,
- Radio programs,
- Compilations.

In the framework of Cuidado we are currently exploiting the web with a crawler specifically designed for this task.

4.2.1. The Cuidado Crawler:

It is a multi-thread software designed to crawl the web. Its goal is to gather as many web pages as possible, parsing every word and every link on each page. Each crawled web page is given a score according to the presence of keywords. Each URL gathered on the page is given the score of the page. Several crawling modes are available from blind crawling (no keywords, only a few starting URLs) to narrow crawling (specific keywords that can be changed dynamically)
The Cuidado Crawler can create/handle several crawling database. Each user can create as many databases as his hard drive can contain. Therefore, users can create database on specific topics or according to specific tastes. For example, if you interested in "intelligent techno". There is over 118000 hits in Google[6] for this query and probably more when you will read this. You can start crawling using the first answers provided by Google as well as specific keywords you entered like "new, research, noise, click and avant-garde". Therefore you construct an "intelligent-techno" oriented database

6. http://www.google.com

which favours your vision of intelligent techno thanks to the keywords.

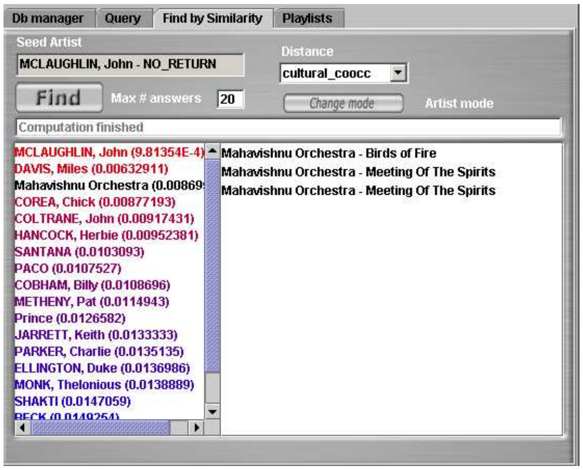

Figure 6 The similarity panel showing artists culturally similar to jazz guitarist John McLaughlin

The second part of this software is devoted to the distance computation. The various formula used here were introduced in [14]. We are looking for occurrences of words in the same page, taking into account the number of pages where each word is found.

4.2.2. Integration in the Music Browser :

To ensure the compatibility with the Music browser users can import any data coming from Cuidado tables. The distance is then computed for each entry and is exported back to the Music Browser as a new distance table. Figure 6 shows the results of a cultural similarity query on the jazz guitarist John McLaughlin. The closest artists include Miles Davis (McLaughlin played on two of his records, “In a Silent Way” and “Bitches Brew” in 1969), the Mahavishnu Orchestra (a fusion band formed by McLaughlin in 1971, including drummer Billy Cobham, also present on the list), jazz pianist Chick Corea (who played with McLaughlin and Miles Davis in the 1969 records), jazz guitarists Pat Metheny and Paco de Lucia (who McLaughlin played in trio with), etc…

5. EXPLOITATION

We have covered so far the core technologies for producing content descriptions of music titles. A key issue is the exploitation of these information on the user side. The graphical interface issue is problematic because of the great variety of behaviours of users, and because the actual devices that will be used for large scale access to music catalogues are still unknown (computers ? set-top boxes ? PDAs ? telephones ? Hard-disk Hi-FI ?). Many user interfaces have been proposed for music access systems, from straightforward feature-based search systems to innovative graphical representations of play lists. For instance, Gigabeat[7] display music titles in spirals to reflect similarity relations titles entertain with each other. The gravitational model of SmarTuner[8], represent titles as mercury balls moving graciously on the screen, to or from “attractors” representing the descriptors selected by the user. The IBM GlassEngine[9] proposes to browse a collection of pieces by minimalist composer Philip Glass, using a set of sliding cursors which rearrange the collection according to several criteria simultaneously (joy, sorrow, density, velocity, etc.). However gracious, these interfaces impose a fixed interaction model, and assume a constant attitude of users regarding exploration: either non-explorative - music databases in which you get exactly what you query - or very exploratory. But the users may not choose between the two, even less adjust this dimension to their wish. The current interface of the Music Browser aims at allowing users to choose between many modes of music access: explorative, precise, focused or hazardous.

5.1. Focused interfaces

The query panel (figure 7) is mostly dedicated to focused search in the database. In this panel users can issue queries on all available artists and songs metadata. These metadata can be editorial: artists’ names, songs’ names, voice quality, etc. as well as computed: subjective energy, tempo, etc. The result of a query is a music titles list. Then this result set can be further filtered to return only songs with fast tempo, or only songs with a male singer. This result list can be transferred to the player for listening/exporting purpose

Figure 7 - Screenshot of the query panel in the Music Browser

---

7. http://www.gigabeat.com
8. http://www.mzz.com
9. http://www.philipglass.com/glassengine

5.2. Explorative interfaces

5.2.1. Sliding between similarities

An interesting issue resulting from the studies on feature-based and cultural similarities is the comparison between these different sorts of similarity. For instance in Figure 5, a starting title such as “Le moment de vérité” played by Ahmad Jamal, is considered by the MB as similar timbre-wise to “Humoresque Op. 20” by Schumann or “Blue and sentimental” by Hank Jones, but culturally, it is closer to “Ahmad’s blues” by Miles Davis, because of the strong relationship between these two players, captured by the web crawler. Of course, there is no grounded truth here, and all these similarities are relevant. The next issue to solve is to aggregate these similarities, or at least propose users simple and meaningful ways of exploiting these different techniques.

In [2], we have proposed an interface, the “aha slider”, which allows the user to rank the results of a query according to two possibly orthogonal types of similarity. The slider is simply a way to filter the result set of one similarity according to the values of the second similarity measure. For instance, one can ask for “timbrally” similar songs which are also very close according to cultural similarity (e.g. “Ahmad’s blues” by Miles Davis), or, on the contrary, filter the result set so that it only contains songs which are culturally very distant from the query (e.g. Schumann or William Sheller).

This interface attempts to give the user full control over the degree of surprise and freedom in the way the system satisfies his request. A non-exploratory behavior (e.g. culturally similar) implies that the system should return exactly the answer to the query, or an answer that is as expected as possible (same title, same artist). An exploratory behavior (e.g. culturally distant) consists in letting the system try different regions of the catalogue rather that strictly match the query.

5.2.2. Playlist Generation

An original feature introduced by the Browser is a powerful playlist generation system, based on constraint satisfaction techniques ([5]). This technique allows user to get entire music playlists from a catalogue, by specifying only abstract properties on the playlist, such as:

- the playlist should contain 12 different titles,
- the playlist should not last more than 76 minutes,
- the genre of a title should be *close* to the genre of the next title,
- the playlist should contain at least 60% of *instrumental* titles,
- the sequence should contain titles with increasing tempo, etc.

The problem of generating such playlists given a very large title catalogue with musical metadata, and a set of arbitrary constraints is a NP-hard combinatorial problem. Moreover, in the case of a contradictory set of constraints, there may not be an exact solution. An ideal system should therefore be able to generate good approximate compromises. The Cuidado Music Browser is able to generate such playlists automatically (figure 8), using a fast algorithm based on adaptive search [5].

We give here an example of a 5-title playlist with the following constraints:

1- Timbre continuity: the playlist should be “timbrally” homogeneous, and shouldn’t contain abrupt changes of textures.

2- Genre Cardinality: the playlist should contain 30% of Rock pieces, 30% of Folk, and 30% of Pop

3- Genre Distribution: the titles of the same genre should be as separated as possible.

One solution found by the system is the following playlist:

- Rolling Stones – You Can’t always get what you want - Genre = Pop/Blues
- Nick Drake - One of these things first - Genre = Folk/Pop
- Radiohead - Motion Picture Soundtrack - Genre = Rock/Brit
- The Beatles - Mother Nature's Son - Genre = Pop/Brit
- Tracy Chapman - Talkin' about a Revolution - Genre = Folk/Pop

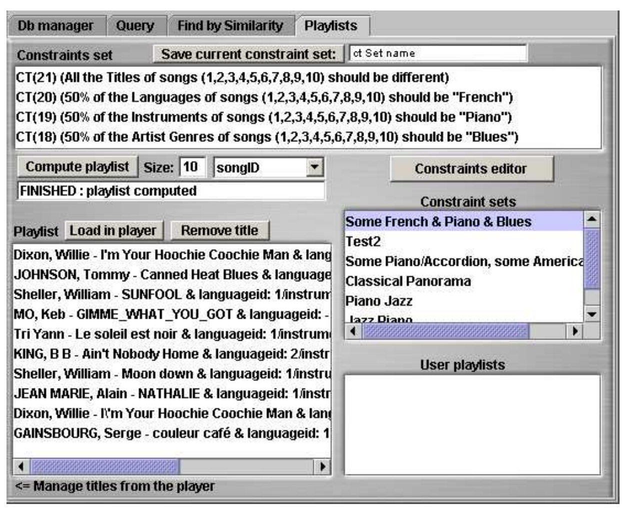

Figure 8 - Screenshot of the playlist generation system

Our current research regarding playlist generation aims at designing simple user interfaces to specify arbitrary constraints in a more intuitive way than in the current implementation, which based on a crude mix of lists and multiple choices. A possible direction towards this is the use of simple drawings or gestures as a way to describe dynamical behaviours (“increasing”), or distribution properties (“a lot of”, “from here … to here”).

6. ARCHITECTURE

This section describes the general architecture of the Music Browser (Figure 9). The central element of the architecture is the metadata server. This server is a MySQL database hosted on a SQL server. The server acts both as a server for PHP scripts and servlets. The MusicBrowser is implemented in Java and communicates with the MySQL database using JDBC drivers. The metadata server runs a PHP server accessible over the Internet. Specific PHP scripts allow client applications to fetch and submit metadata to this server.

The music browser contains four panels aimed at music title access: the player, the query panel, the similarity panel and the playlist panel.

Additionally, the browser includes two management tools: the editorial data management tool and the extractor and computation management tool. The purpose of the computation management tool is to compute descriptors for the songs in the database as well as similarity measures. It can use any stand-alone extractor (exe or bat files) developed by third party.

The editorial metadata management tool is used to manage artists and songs properties. It provides choice lists for each property and enables basic editions such as title name or keywords, as well as title genre, primary and secondary artist, as described in section 2.1. This tool interacts on-line with our metadata server.

Lastly, with the apparition of ad-hoc networks, users can share their data easily with other users and in a transparent way. This situation raises an issue in the management and synchronization of metadata. We describe in [11] a solution to allow both private and shared metadata to coexist in a single environment.

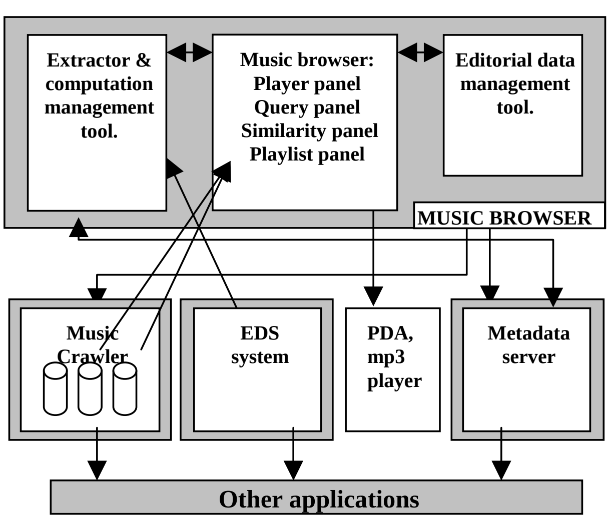

Figure 9 - Interaction between the different components of the Music Browser

7. FUTURE WORK : TOWARDS AN API

Our experience in designing a large–scale EMD system such as the Music Browser shows that the main difficulty is to combine several systems/languages in a seamless manner : a database (SQL), an object-oriented engine to manage “multimedia items”, like songs, artists, albums, etc. (JAVA), user interfaces/interaction modules (JAVA), signal processing algorithms and extractors (Matlab, C), music rendering (JMF). All of these aspects interoperate closely, e.g. the interface calls an executable which computes a value, which is stored in the db, and re-used in another interaction module.

This architecture, although it does not present any particular technical difficulty, is expensive to design, and requires much incremental “doodling” both to specify and to build. On the other hand, such an architecture is needed for many other applications than the Music Browser, virtually every application concerned with content-based interaction, access, browsing of large multimedia collections. Among other Sony CSL projects, the Musaicing ([23]), a composition tool to create sequences of samples according to high-level properties on their metadata (e.g. a steady tempo, with some voice samples, a given energy profile, etc.), and Personal Radio ([13]), an automatic, customized radio station, are based on the same type of architecture.

Moreover, the overhead of building such an architecture is often a limiting factor for many subtasks like evaluating content-extraction algorithms, a problem which is hotly debated in the music information retrieval community ([9]). As described in [3], in order to evaluate and fine–tune algorithms like the timbre similarity used in the Music Browser, one needs to be able to :

- access and manage the collection of music signals the measures should be tested on
- store each result for each song (or rather each duplet of songs as we are dealing with a binary operation dist(a,b)=d) and each set of parameters
- compare results to a ground truth, which should also be stored
- build or import this ground truth on the collection of songs according to some criteria
- easily specify the computation of different measures, and to specify different parameters for each algorithm variant, etc...

Following these experiments, we have started developing a more general API, the so-called MCM (multimedia content management). MCM is a set of java classes, which offer the following data structures and functionalities :

- multimedia *items* (e.g. songs or artists), existing synchronously both in db and in memory.
- *fields* or metadata for each of these items (e.g. song's tempo or artist's name).
- *field values* for each item are read/written in db, and can be cached in memory for applications which require more CPU power, like playlist generation.
- items may link one to another (e.g. song items can be associated with artist items, video clip items, etc.). These associations are treated like fields (the "artist" item is a metadata of the "song" item), which values are the corresponding items.
- some fields are computable, i.e. their value is the output of an extractor, either computed online or offline, in batch mode.
- items can link to other items with *relations*, e.g. timbre or cultural similarity.
- items, fields, relations can be added (e.g. add a new directory of mp3s in the Browser, add a third-party extractor, etc.), updated, retrieved or deleted from the db.

Using MCM, all the architectural difficulties of creating databases, synchronizing data, calling extractors are hidden out. Applications like the Music Browser can be developed very quickly, by concentrating only on meaningful, higher-level concepts. Like for the EDS, we think that this is a potentially important contribution to the Music Information Retrieval community as it is a first step towards a unified vision of content based interaction and access systems.

8. CONCLUSION

The Cuidado music browser is the first large scale, fully content-based music access system. It includes all the technologies needed to extract descriptors, create similarity relations, and make these information easily available to users. The system is fully operational, and user tests have started to assess the usability of content information for music access. Two side projects emerged from the design of this system : the EDS, a general framework for the automatic design of extractors, and MCM, an API to speed up the design of applications concerned with extracting and exploiting musical metadata for browsing music. Both projects constitute a first step towards a unified vision of content based interaction and access systems.

9. REFERENCES

[1] Allamanche, E. Herre, J. Helmuth, O. Frba, B. Kasten, T. and Cremer, M. (2001) "Content-Based Identification of Audio Material Using MPEG-7 Low Level Description" in Proc. of the 2nd International Symposium on Music Information Retrieval. (ISMIR 01), Bloomington, Indiana, USA.

[2] Aucouturier, J.-J., Pachet, F., Sandler, M. (2004) « The way it sounds : Timbre models for structural analysis and retrieval of music signals", submitted to IEEE Transactions on Multimedia, 2004.

[3] Aucouturier, J-J, Pachet, F. (2004) Improving Timbre Similarity : How high's the sky ?, submitted to Journal of Negative Results in Speech and Audio Sciences (JNRSAS), 2004.

[4] Aucouturier, J.-J. Pachet, F. (2003) Musical Genre: a Survey, In Journal of New Music Research, 32:1, 2003.

[5] Aucouturier, J.-J. Pachet, F. (2002) Scaling up Playlist generation, In Proc. of the IEEE International Conference on Multimedia and Expo (ICME 02), Lauzanne, Switzerland.

[6] Belkin, N. (2000) Helping people find what they don't know. In Communications of the *ACM* Vol. 43, N. 8, August 2000, pp. 58-61.

[7] Berenzweig, A. and Ellis, D. (2001) Locating Singing Voice Segments within Music Signals. in proc. IEEE Workshop on Applications of Signal Processing to Acoustics and Audio (WASPAA 01), Mohonk, NY, USA.

[8] Cohen, W., Fan, W. (2000) Web-Collaborative Filtering: Recommending Music by Crawling The Web, in proc. 9th International World Wide Web Conference (WWW9), Amsterdam, The Netherlands.

[9] Downie, S. (2003). Toward the scientific evaluation of music information retrieval systems. In proc. International Symposium on Music Information Retrieval (ISMIR 03), Baltimore, Maryland, USA.

[10] Herrera, P, Serra, X. Peeters, G. (1999). Audio descriptors and descriptors schemes in the context of MPEG-7. Proceedings of the International Computer Music Conference (ICMC 99), Beijing, China.

[11] La Burthe A., Pachet F., Aucouturier JJ. (2003) Editorial Metadata in the Cuidado Music Browser: between universalism and autism. In proc. 3rd International Conference of Web Delivering of Music (WedelMusic 03), Leeds, UK.

[12] Pachet, F. and Zils, A. Evolving Automatically High-Level Music Descriptors From Acoustic Signals. Springer Verlag LNCS, 2771, 2003.

[13] Pachet F. (2003) Content Management for Electronic Music Distribution: The Real Issues. Communications of the ACM 2003.

[14] Pachet, F. Westerman, G. Laigre, D. (2001) Musical Data Mining for Electronic Music Distribution., Proceedings of First International Conference of Web Delivering of Music (WedelMusic 01), Firenze, Italy.

[15] Pachet F., Cazaly D. (2000). A taxonomy of musical genres. In Proc. Content-based Multimedia Information Access (RIAO), Paris, France.

[16] Peeters, G. Rodet, X. (2002) Automatically selecting signal descriptors for sound classification. Proceedings of the International Computer Music Conference (ICMC 02), Goteborg (Sweden).

[17] Scheirer, Eric D. (1998) "Tempo and beat analysis of acoustic musical signals", in Journal of the Acoustics Society of America. (JASA) 103:1 (Jan 1998), pp 588-601.

[18] Scheirer, Eric and Slaney, Malcolm. Construction and evaluation of a robust multifeature speech/music discriminator. In proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 97), Munich, Germany.

[19] Tzanetakis, George and Perry Cook (2002) "Musical Genre Classification of Audio Signals", IEEE Transactions on Speech and Audio Processing, 10(5), July 2002

[20] Wold, E. Blum, T. Keislar, D. Wheaton, J. (1996) Content-Based Classification, Search, and Retrieval of Audio, in IEEE Multimedia, 3:3, pp. 27-36.

[21] Zils, A. & Pachet, F. Extracting Automatically the Perceived Intensity of Music Titles. Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFX03), London, UK.

[22] Zils A., Pachet F., Delerue O., Gouyon F. (2002) Automatic Extraction of Drum Tracks from Polyphonic Music Signals. Proc. 2nd International Conference of Web Delivering of Music (WedelMusic 02), Darmstadt, Germany.

[23] Zils, A. and Pachet, F. (2001) Musical Mosaicing, Proceedings of COST-G6 Conference on Digital Audio Effects (DAFX01), Limerick, Ireland.