The CUIDADO Project

Hugues Vinet
IRCAM
1, place Igor Stravinsky
F-75004 Paris, FRANCE
vinet@ircam.fr

Perfecto Herrera
IUA-Universitat Pompeu Fabra
Pg. Circumval·lació, 8
08003 Barcelona, SPAIN
pherrera@iua.upf.es

François Pachet
SONY-CSL
6, rue Amyot
F-75005 Paris, FRANCE
pachet@csl.sony.fr

ABSTRACT

The CUIDADO Project (Content-based Unified Interfaces and Descriptors for Audio/music Databases available Online) aims at developing a new chain of applications through the use of audio/music content descriptors, in the spirit of the MPEG-7 standard. The project includes the design of appropriate description structures, the development of extractors for deriving high-level information from audio signals, and the design and implementation of two applications: the Sound Palette and the Music Browser. These applications include new features, which systematically exploit high-level descriptors and provide users with content-based access to large catalogues of audio/music material. The Sound Palette focuses on audio samples and targets professional users, whereas the Music Browser addresses a broader user target through the management of Popular music titles. After a presentation of the project objectives and methodology, we describe the original features of the two applications based on the systematic use of descriptors and the technical architecture framework on which they rely.

1. OBJECTIVES AND METHODOLOGY

The CUIDADO project (Content-based Unified Interfaces and Descriptors for Audio/music Databases available Online) aims at delivering new applications based on the systematic use of audio and music descriptors, which address new needs of the professional and consumer audio/music chain. This approach falls within the scope of the MPEG-7 standardization process, in which CUIDADO participants have been already actively involved [21]. In this framework, descriptors and descriptor schemes combine knowledge bases with numerical features of the audio contents, and are designed for several classes of applications. First, as metadata, enabling content-based retrieval functions on large audio/music databases, which would not be feasible, due to data volumes, by directly accessing audio samples. Secondly, for allowing users to manipulate audio/music contents through high-level specification, designed to match the human cognitive structures involved in auditory perception, as opposed to traditional low-level, implementation-driven data structures. The IRCAM Spatialisateur project, where perceptual parameters are used for controlling the sound quality of a simulated room effect, has proved the interest and usability of such an approach [12]. Similarly, mixing constraints have been introduced as a particular form of content descriptors allowing real time adaptation of complex music mixings [17]. More generally, the design of appropriate descriptors makes possible new application models for accessing and manipulating audio contents and represents a major opportunity for the development of the music production and distribution industries.

However, in order to come up to effective results, several constraints, which form the basic methodological framework of the project, must be taken into account:

i) the necessity of a top-down approach, aiming at deriving high-level descriptors, which form the basic knowledge structures accessible by the user in target applications. These knowledge structures must be adapted to application functions and consistent with the user cognition, i.e. the way his own knowledge about sound and music is organized.

ii) the interest in combining this top-down approach with a bottom-up approach, which extracts low-level descriptors from an automatic analysis of the audio signal. The objective is to automatically compute required high-level descriptors, avoiding manual input when possible. The main related issues, discussed in [10, 11] in the context of sound classification, are how to *choose* a set of relevant low-level descriptors and how to *map* targeted high-level descriptors onto them, by using appropriate machine learning techniques.

iii) designing and validating descriptors in laboratory conditions may however not suffice for real-world applications. In order to enable a complete assessment procedure, it is necessary to build fully functional application prototypes, which take into account all technical constraints, and handle man-machine interaction issues. Machine learning functions enable the system to learn the user's knowledge structures; symmetrically, the design of the user interface is a key factor for enabling the user to assimilate the system knowledge structures.

iv) such an ambitious program requires the combination of a multi-disciplinary scientific and technological expertise, which has been gathered for the CUIDADO project in an international consortium including approximately 40 researchers and engineers from IRCAM (coordinator), IUA-UPF, SONY-CSL, Ben Gurion University, and industrial partners (Oracle, Creamware, Artspages). The project is supported by a grant from the European Commission (Information Society Technologies Program).

In order to fulfill these objectives, the project is organized around the development of two target applications, the Sound Palette, and the Music Browser, which address different needs. These applications are presented in the next sections.

2. THE SOUND PALETTE

The Sound Palette is dedicated to professional users in music and audio production. It offers audio sample management and editing features based on sound content description.

2.1 Sample management features

The notion of audio sample is taken here in its usual denomination, i.e. sounds of short duration (typically less than 20 seconds), which contain individual events, such as single instrument notes. Actually, the system can handle samples with more complex contents, but the main assumption made on related descriptors is that they are computed for the whole sound considered as a single entity, even if they reflect its internal structure, e.g. as a decomposition into different temporal phases.

The first available application using content-based descriptions for sample management has been SoundFisher by MuscleFish [13]. The Studio Online project developed at IRCAM [3][25] provided an online server giving access to a 117,000 instrument sample database, where high-level retrieval functions were experimented: browsing through a specifically designed hierarchical taxonomy of samples (including contemporary playing modes) and a query by example interface, relying on an elementary perceptual similarity distance resulting from studies on timbre spaces. A more general descriptor scheme for instrument timbre was proposed and integrated in the MPEG-7 standard [16].

Targeted features for the Sound Palette include various enhancements, in particular by enabling the management of the user's own sounds, and not only access to a fixed database as in Studio Online.

They combine the following functions:

- management of a global sample repository, shared among several users, containing both a reference set and user samples, and available as a set of online Intranet/Internet services.

- classification tools, which help the user to organize his sounds in classes, and provide automatic classification solutions when introducing new sounds. Classes can be learned from arbitrary user categories (as long as they refer to the sound content), and can also be trained from more objective classification schemes, such as sound sources, or morphological characteristics (such as pitch or amplitude envelopes, grain, etc.).

- sample retrieval functions, combining criteria used for classification (limitation of the search set) and query by example functions, based on similarity distances, computed between each pair of samples across class boundaries.

The first tests of automatic classification algorithms, based on static low-level descriptors (computed for the whole sound duration), performed on standard instrument classes, show promising results [22]. New, more robust, perceptual similarity functions are also proposed, based on recent results of a meta-analysis of main existing studies on timbre spaces [15]. Further developments foresee the use of dynamic descriptors, which model the evolution of various sound features over time, and the extension of proposed search functions through the use of textual attributes, which qualify various aspects of the sound contents.

All these features combine low-level descriptors automatically computed from audio signals and high-level descriptors, which are either provided by the system or user-defined. The design principle for low-level descriptors is to set up an as comprehensive set as possible, including characteristics which account for the global temporal evolution, spectrum shape, energy, harmonicity, timbre dimensions, etc. Depending on targeted high-level features (e.g. for learning the characteristics of a given classification scheme), the most relevant low-level descriptors are automatically selected from this redundant low-level descriptor set using selection criteria such as the computation of mutual information between descriptors and classes [22].

Two versions of the Sound Palette are developed in parallel:

- an "online version", which includes all above functions, and is experimented as a common sample database for all musical productions at IRCAM.

- an "offline version", integrated in an audio production workstation such as the Creamware Pulsar system, which is restricted to the management of the user's samples on his local PC, but also includes sample editing features described in the next section.

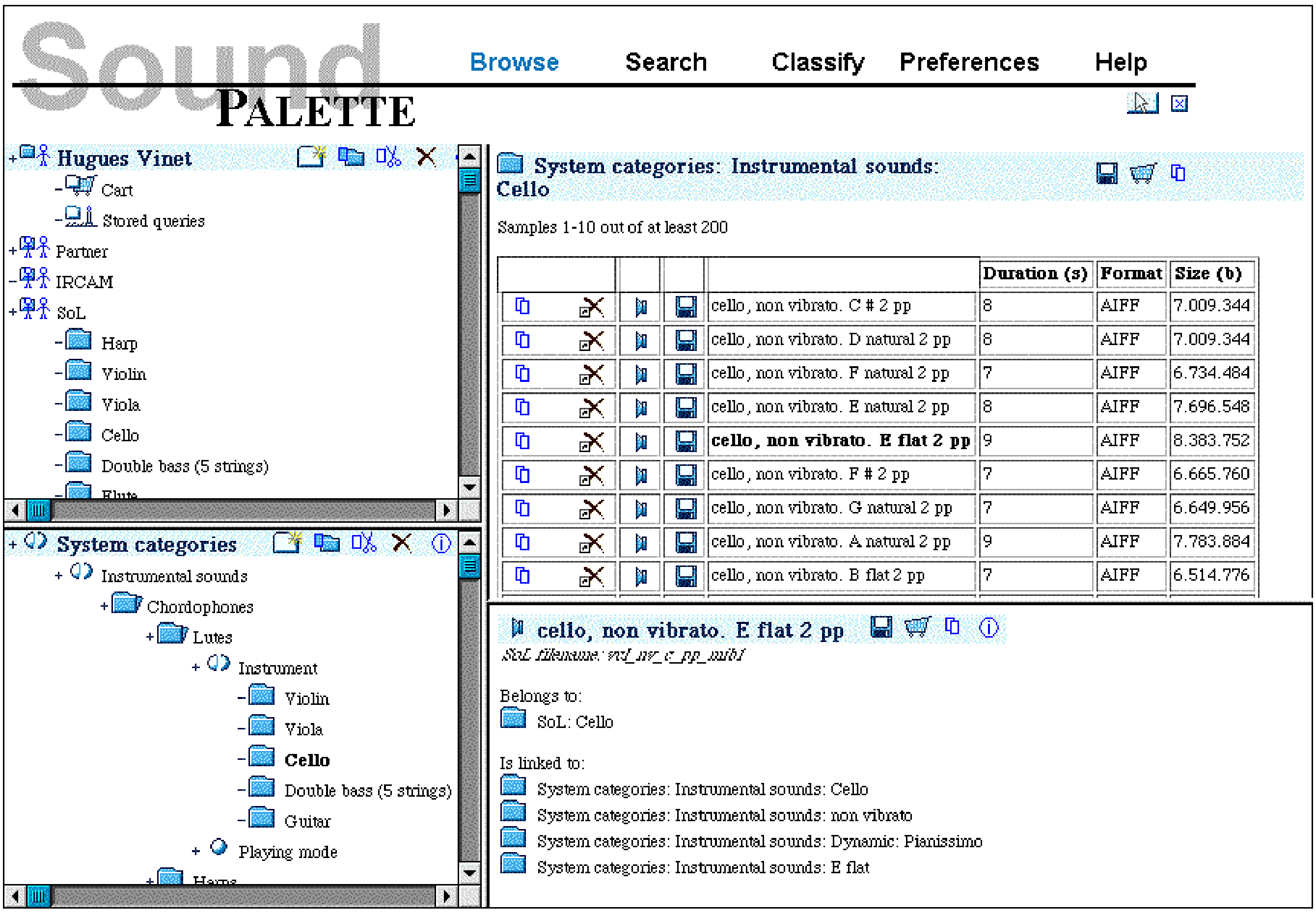

Figure 1. Browse screen of the Online Sound Palette

Figure 1 shows the "Browse" screen of the Online Sound Palette, whose interface mainly relies on HTML and Javascript protocols and can be accessible from any Web browser. The top-left frame is an editor for personal user data: user-created sample directories, reference sample collection directories he has access to, etc. The bottom-left frame is dedicated to high-level description schemes, such as a taxonomy of sound sources, or any other classification scheme. The top-right frame displays the list of items from the selected directory (in that case, the Cello directory in the System category), with the possibility of listening to (by streaming) and downloading any sound or group of sounds. Multiple selection from various directories is available. Cut-copy-paste functions between directories are also available, but, unlike file management system metaphors, a sound can be linked to any number of directories. The bottom-right frame displays details of the selected sample on the upper list, with all directories it is related to. Another screen enables a multi-criteria search, including full text search using sound names, directory names, and free user textual comments on sounds and directories. The system features a complete management of access rights (see, listen, modify/delete) for samples and directories. This application is designed and developed at IRCAM by the Online Services Team .

Figure 2 shows the "Sound Palette Browser" window of the Offline Sound Palette, integrated by Creamware in its Pulsar system, which displays similar information areas, such as sample categories and lists, as well as the waveform of the selected sample.

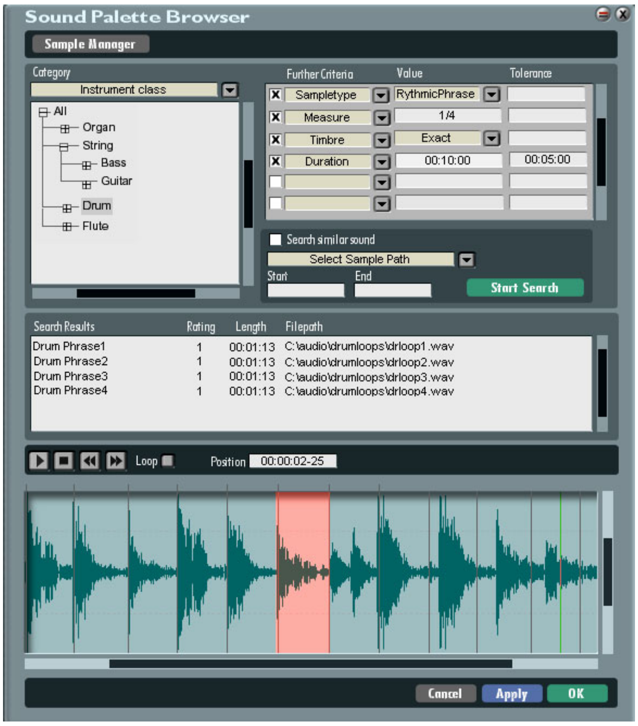

Figure 2. File browsing and retrieval sections of the Offline Sound Palette

2.2 Sound editing features

The Sound Palette can handle three different kinds of sound materials: sound samples, music monotimbral phrases, and rhythm tracks (for example the "drum loops" that are ubiquitous in popular music).

Deriving a multi-level music description of these materials and populating large databases with them paves the way for new navigation, editing and transformation possibilities. The functionalities are even more enhanced when a synthesis engine is allowed to play some role in the general architecture of the system. Provided this system, we can talk about "content-based sound edition", a concept that has much in common with old proposals of "intelligent sound editors" [4]. More recent research on which the Sound Palette (offline version) relies can be found in [1, 5, 6, 7, 9, 11, 14, 24]. It should also be noted that nowadays there is a bunch of commercial applications that implement content-related functionalities, though in most of the cases, content representation is hidden to the user. We could mention some audio-to-MIDI programs, such as Wildcat's Canyon's *Autoscore*, Epinoisis's *Digital Ear*, or Recognisoft's *SoloExplorer* that perform some rough transcription, even attempting instrument recognition. These tools are not aimed, though, to do more than MIDI approximations of some audio content, and their performance is, to put it politely, not optimal. Propellerhead's *ReCycle!*, Digidesign's *Beat Detective*, and Sonic Foundry's *Acid* are tools for loop manipulation that use some segmentation tricks and rhythm description in order to re-arrange or re-create a music file. Those "descriptions" are stored in proprietary file formats. Another conceptually related piece of software is Celemony's *Melodyne*, a tool for flexible transformation of audio that allows the user to transpose a musical phrase or some of its individual notes without timbre corruption, to stretch or compress notes along the time axis preserving their attacks unchanged, to change vibrato and articulation inflexions, and finally deriving a MIDI representation of the musical content. As we are going to see, there are some editing and transformation functionalities (provided by the types and levels of music content representation they manage) that none of them are currently exploiting. Navigation in the Sound Palette editing environment may use note, phrase or instrument labels instead of the handwritten markers that are usually mandatory in standard editors. "Skip to next note" or "Skip to next snare hit" provide better location than spotting the cursor somewhere "there" by visually inspecting the waveform and scrub-listening to it.

Automatic segmentation and labeling of content also allows multiple selections of those segments. Once they are selected, it is possible to perform editorial actions on all of them simultaneously. For example, after automatically describing an electronic bass phrase with note and sub-note descriptors, it is possible to select a portion of their attacks and chop them (a well-known technique in certain popular styles). This is done in a compact way instead of manual-visual selection and change of each one of the portions to be edited.

Creation of MIDI maps from an audio rhythm loop is another convenient function that cannot be completely achieved with current commercial software devoted to this type of sonic materials (e.g. Recycle!). A Sound Palette rhythm description contains not only information about onset location of sounds and basic pulses that underlie in the track but also generic labels for the percussion instruments. This way, a MIDI "recreation" of an example drum loop can be set up for using our owned drum kit samples, instead of some illegal CD-lifted fragment. Thanks to a perceptual similarity-based search function, drum samples used in the recreation can even be perceptually very close to the original ones (provided a large sound database, indeed!), and not only the same in terms of category.

As editing and transforming sounds become operations with no clear boundaries between them, transformation comes into play even in editorial tasks such as concatenating two rhythmic patterns for using the composite as a song building block. Let's suppose that pattern A has a global tempo of 100

Figure 3. Edition and transformation windows of the Offline Sound Palette.

The upper section contains content-related editing functions and time-scale transformation functions. The lower one contains "loop reconstruction" and some event-based functionalities.

BPM and pattern B scores at 80 BPM, and we want the composite to be at 93 BPM. Rhythm tempo matching can be achieved by conveniently time-stretching the original patterns. Seamless and continuous fusion between different-tempo rhythm patterns is possible without bothering the user with tempo or stretch parameter adjustments.

Variation generation is a must in the composer armamentarium, and the Sound Palette implements some options for that. Traditional variations on a musical phrase by tonality change of a phrase are achieved through transposition of specific notes from the original phrase. Rhythm tracks can also be altered by swapping a specified instrument by another, or by muting the specified one.

More original variations can be achieved by applying a "musaicing" process to a musical phrase as described in next section [26]. Provided that we get access to a sound database of "world music" voices, a vocal track could be transformed into a timbral mosaic containing bits of those voices (instead of the original one) though keeping the same pitch than the original.

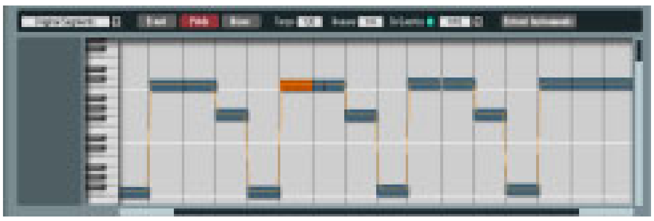

Figure 4. Pitched events editing section of the Offline Sound Palette

3. THE MUSIC BROWSER

The Music Browser is intended to be the first content-based music management tool for large music catalogues. This application is targeted to all the actors of the music distribution industry: labels who want to exploit systematically their catalogues, music distributors who want to propose customers with personalized music playlists, radios who have to schedule music programs that match user's tastes, copyright societies, who want to control illegal copies of music titles, and more generally the new businesses of Electronic Music Distribution yet to come.

Several music browsers have been made available to the public recently. Most of them are title-based, that is: the user has to explicitly state what he is looking for. The few ones that propose some sort of content-based approach (MoodLogic, Relatable) suffer from two main drawbacks:

1) The used metadata is entirely manual. MoodLogic accumulated a huge database of music descriptors, built from collaborative manual inputs. Manual metadata is however very expensive to build and maintain. Besides, high-level descriptors are usually hardly consensual (what does "Rock" mean?).

2) There is no facility for building automatically consistent music playlists. Titles are retrieved individually, forcing the user to organize his/her own playlists, which is time-consuming, and requires a good knowledge of music classifications.

The aim of the CUIDADO Music Browser is two-fold. Technically the goal is to implement the whole processing chain, from the music data itself (signal, Midi, or reference information) to the user, with as much automatic processing as possible, and with a flexible music playlist facility. Conceptually, the aim of the Music Browser is to build evidence that content-based music management does allow efficient personalized music access and brings substantial added value in comparison to existing music distribution schemes.

Technically, the main challenge of such a music browser is to be useful for the largest possible set of music behaviors. Preliminary experiments show indeed that user behaviors exposed to large music catalogue may be extremely different. Some users may want to browse using top down genre categories, others are similarity and example oriented. Some know exactly what they look for, others want explicitly only new, unknown titles, etc [20]. The Music Browser consequently implements all the browsing mechanisms, including:

- *Descriptor-based* search: "I want titles with fast tempo, high energy, in the Rock generic genre",

- *Similarity-based* search: "I want titles which are "close" to a given set of titles". The similarity may be picked up from a list of several similarity relations, each one with a different semantics, e.g. cultural similarities, metaphoric similarities, artist-based similarities, etc

- *Global search*: "I want a playlist with instrumental music at the beginning and dance music at the end, with increasing tempo values, and more than 60% of female singing voice".

Navigation within titles is also considered, through the use of automatically computed music summaries, which index the main title parts through a newly designed segmentation method [23].

The most important modules of the processing chain, illustrated in Figure 5, are presented hereinafter.

3.1 Unary descriptor extractor module

This module implements top-down signal processing techniques that extract global unary information about music titles. Music titles are considered here as 3 to 4 minute arbitrary complex music signals. The descriptors considered target the music title in its entirety, as opposed to low-level, local descriptors that consider only small portions of a signal (as in the Sound Palette application).

Unary descriptors are interesting for two reasons. First, as such, they provide a basis for descriptor-based queries. Second, they yield similarity relations that can be exploited during music search.

More precisely, the descriptors targeted are of three sorts:

1) *Rhythm descriptors*. This category includes tempo/beat, as well as rhythmic information. Preliminary results about rhythm extraction have been obtained and documented in [8]. In particular, we have focused on percussive rhythm, i.e. rhythm produced by repeated occurrences of percussive sounds.

2) *Energy-based descriptors*. These descriptors aim at extracting rough categories of perceptual energy. A typical example is the intuitive perceptive difference between a hard rock piece and an acoustic guitar folk song. The mapping between perceptual energy and low-level descriptors is done using machine learning techniques applied on a vector of descriptors such as physical energy, energy variance, tempo, etc.

3) *Timbre-based descriptors*. These descriptors aim at extracting a global representation of the overall timbre of the song. The resulting descriptor is not meaningful in itself, but serves as a basis for establishing timbre similarities between different songs.

3.2 Similarity analysis modules

Low-level descriptors produce low-level similarities. There are other means of extracting higher-level similarity relations for music. A well-known technique for producing similarities is *collaborative filtering*. This technique builds similarities between titles based on similarities between user profiles. Although the technique does establish relevant music similarities, it carries with it a number of important drawbacks, which limit in practice its usability, such as the need for an initial, large, set of user profiles, and the degradation of quality in recommendations for rich, eclectic profiles. We have investigated other techniques based on data mining, applied to corpuses of textual information. The studied corpuses range from collections of album playlists (e.g. as found in CDDB), radio program listings, and general search engines (e.g. Google). The techniques used are based on co-occurrence, i.e. each time two titles are found in a similar context (radio program, album playlist or web page), their co-occurrence counter is incremented. Eventually we build a similarity measure from this co-occurrence matrix. We have shown in [18] that the extracted similarity is meaningful, non-trivial, and complementary to the similarities extracted through low-level descriptors.

The extracted similarity relations are then exploited in the music browser for similarity-based search, as well as for proposing music discovery schemes. Through a simple user interface, the user can quickly explore regions of the catalogue which are either very close to his/her set of preferred titles, or very far (see [19] for a prototype implementing varying-length music catalogue exploration).

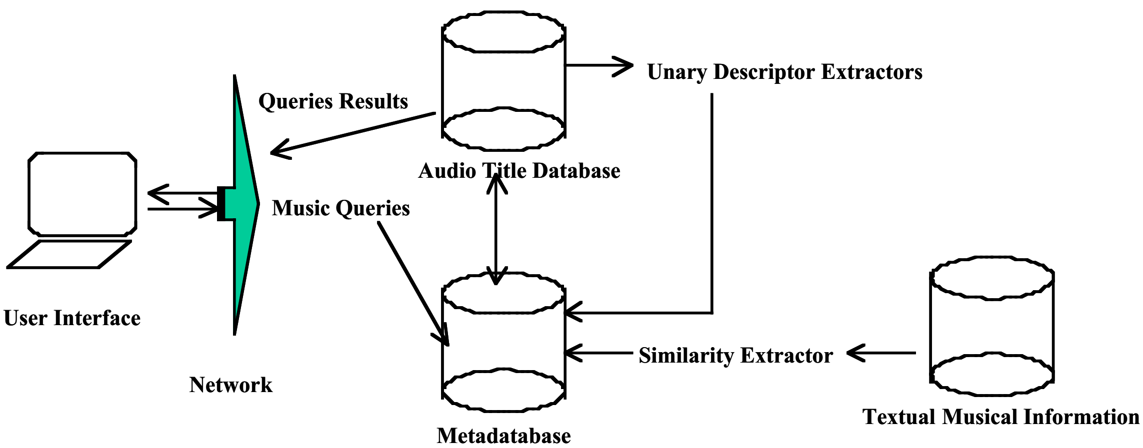

Figure 5. The main content-management modules of the Music Browser

3.3 Playlist generation module

As mentioned in the preceding section, music titles are rarely accessed and listened to in isolation. We have proposed in [18] to consider sequence generation as the main paradigm for music retrieval, instead of the standard title-based approach. In particular, sequence generation allows user to explore efficiently the catalogue without having to learn the underlying descriptor ontologies. The main idea is to let the user express global properties of music playlists, such as continuity (I want a music playlist which is continuous, tempo-wise, or style-wise, etc.), cardinality (at least 60% instrumental, Funk titles), and distribution (no repetition of more than two titles by the same artist). It was shown that specifying properties of playlists is indeed much easier than specifying properties of individual, unknown titles.

For instance, a 10-title playlist can be specified with the following properties:

-All titles are different,

-Increasing tempo,

-Three cardinality constraints on genre:

- o less than 50% World,
- o exactly 4 Rock,
- o more than 50% Folk.

A solution to this play list generation “problem” found from our 20,000 title test catalogue is:

1) Slip slidin' away - Simon and Garfunkel - tempo 63 - genre = Folk \ Pop

2) Un autre monde - Telephone - tempo 71 - genre = Rock \ Song

3) Barbès - FFF - tempo 78 - genre = Rock \ Funk

4) Waiting for the Miracle - Leonard Cohen - tempo 82 - genre = Folk \ Pop

5) I Palami Sou – Angelique Ionatos - tempo 98 - genre = World \ Mediterranean

6) The Sire Of Sorrow - Joni Mitchell - tempo 113 - genre = Folk \ Pop

7) Tom’s Dinner – Suzanne Vega - tempo 123 - genre = Folk \ Pop

8) You're So Great - Blur - tempo 133 - genre = Rock \ Brit

9) The Boy In The Bubble - Paul Simon - tempo 138 - genre = Folk \ Pop

10) The return of the lost palmas 7 - Madness - tempo 144 - genre = Rock \ New Wave \ Ska

However, finding solutions for large catalogues is inherently NP-hard. We have developed a new algorithm for finding quickly solutions for catalogues of up to 200,000 titles [2], and able to handle arbitrary complex properties. This technique is efficient enough to produce solutions in a few seconds, as well as to form a basis for implementing relevance feedback techniques. In this scheme, the user selects a few sequence properties off-the-shelf, gets quickly a solution, and then refines interactively this solution, either by explicitly deleting unwanted music titles, or by adding or removing sequence properties.

The same sequence generation module has also been applied to the problem of generating sound sequences from a catalogue of samples, as designed and implemented in the Sound Palette application. The result, coined as Musaicing [26], is a new generation of audio sequencers, in which the user specifies the global properties of the sequence to generate, instead of choosing and retrieving explicitly individual samples. A preliminary prototype has been implemented, and current work focus on the design of a library of sound sequence properties, adapted to the generation of popular music styles.

3.4 Web Music Monitoring System

This module, integrated in the Music Browser, is specifically intended for copyright societies. It performs the identification of unknown music excerpts, e.g. found on the Web, with a reference copyrighted title database. So this method addresses the issue of music copyright protection, in a different way than watermarking, which requires the insertion of a digital identifier in the audio signal and is more sensitive to modifications introduced by audio coding processes, such as compression. The identification principle is based on signal comparison. It relies on audio signatures, stored as metadata, which capture relevant information by encoding original signals with high compression rates. Tests performed with 12 seconds excerpts on a database of 650 titles give a successful matching rate greater than 97%.

4. ARCHITECTURE

A common hardware and software architecture has been designed and set up for both online applications. It provides all required services, including: massive data storage, audio streaming, file uploading and downloading, audio file, metadata and user data management, middleware, etc. All used protocols are chosen to be as standard as possible. Access to online services is provided from a variety of client terminals, through the use of HTML/XML when possible, otherwise Java protocols for client interfaces. The physical architecture, based on a 3-tier architecture, is shown in Figure 6. Logical functions are split between four Linux servers and a massive storage system, interconnected through two private networks: a control data network for middleware exchanges, and a high bandwidth network for sound file transfers.

All sound files are stored in the massive storage system (from Network Appliances), and all data handled by the database system, including descriptors, are stored in the database server. The sound and processing server performs all required calculations on sound files, including descriptor extraction. This modular architecture minimizes system administration and enables a selective upgrade of servers when necessary.

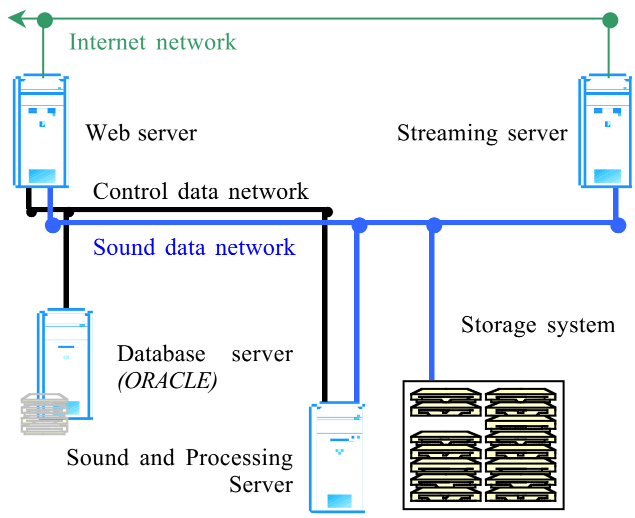

Figure 6. System architecture

5. CONCLUSION

The main objectives of the CUIDADO project have been presented, as well as the design principles of its main target applications: the Sound Palette and the Music Browser. The features of these prototypes illustrate new possibilities in various application fields that directly result from research, performed as part of the project, on audio and music sound description. At the time of this publication, the first versions of the applications are going to be released and tested by selected users. This test and assessment period will be essential for refining the proposed features according to users' reactions, testing and optimizing the system performances in real situations, and converging to the final application development scheduled for the end of 2003. Updated information about the project can be found at http://www.cuidado.mu/ and http://www.ircam.fr/cuidado.

6. ACKNOWLEDGMENTS

The CUIDADO project is funded by the Information Science and Technology Program of the European Commission.

7. REFERENCES

[1] Amatriain, X., J. Bonada, A.Loscos, and X. Serra,. "Spectral Processing", Udo Zölzer (Ed.), DAFX: Digital Audio Effects, John Wiley & Sons Publishers, 2002

[2] Aucouturier, J.-J. and F. Pachet "Scaling up Music Playlist Generation", IEEE International Conference on Multimedia Expo, Lausanne (Switzerland), August 2002.

[3] Ballet G., R. Borghesi, P. Hoffmann, F. Lévy, "Studio Online 3.0: An Internet "Killer Application" for Remote Access to IRCAM Sounds and Processing tools", Proc. of Journées d'Informatique Musicale (JIM'99, 1999), available online at: http://www.ai.univ–paris8.fr/~jim99/actes_html/BalletJIM99.htm

[4] Chafe C., B. Mont-Reynaud and L. Rush (1989) "Toward an Intelligent Editor of Digital Audio: Recognition of Musical Constructs" In C. Roads (ed.), The Music Machine. MIT Press, Cambridge, MA.

[5] Dubnov, S. "Extracting sound objects by independent subspace analysis", Proceedings of AES22 International Conference on Virtual, Synthetic and Entertainment Audio. Espoo, Finland, 2002.

[6] Garcia, D., and X. Amatrian, "XML as a means of control for audio processing, synthesis and analysis", Proceedings of MOSART Workshop on Current Research Directions in Computer Music, Barcelona, 2001.

[7] Gomez, E., F. Gouyon,., P. Herrera, and X. Amatriain, (in preparation). "Music Description Schemes and the current MPEG-7 standard."

[8] Gouyon F, O. Delerue, F. Pachet, "On the use of zero-crossing rate for an application of classification of percussive sounds" Digital Audio Effects Conference, Verona (It), December 2000.

[9] Gouyon, F. , P. Herrera, and P. Cano,"Pulse-dependent analyses of percussive music", Proceedings of AES22 International Conference on Virtual, Synthetic and Entertainment Audio. Espoo, Finland, 2002.

[10] Herrera P., G. Peeters, S. Dubnov, "Automatic Classification of Musical Instrument Sounds", Journal of New Music Research, 2002 (to appear).

[11] Herrera, P., A. Yeterian, and F. Gouyon, "Automatic Classification of Drum Sounds: A comparison of feature selection methods and classification techniques", Proceedings of the 2nd International Conference on Music and Artificial Intelligence, Edimburgh, United Kingdom, 2002.

[12] Jot, J.M., "Efficient Models for Distance and Reverberation Rendering in Computer Music and Virtual Audio Reality." Proceedings of the International Computer Music Conference. San Francisco: ICMA, 1997

[13] Keislar, D., T. Blum, T., J. Wheaton, & E. Wold,. "A content-ware sound browser". Proc. of the International Computer Music Conference, ICMA, 1999.

[14] Klapuri, A., "Sound Onset Detection by Applying Psychoacoustic Knowledge", IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1999.

[15] McAdams S, S. Winsberg, "A meta-analysis of timbre space. I: Multidimensional scaling of group data with common dimensions, specificities, and latent subject classes" Journal of the Acoustical Society of America, (in preparation)

[16] ISO/IEC FCD 15938-4 Information Technology - Multimedia Content Description Interface - Part 4 Audio.

[17] Pachet, F., O. Delerue. "On-The-Fly Multi Track Mixing", preprint of the Audio Engineering Society, 109th Convention, Los Angeles, 2001.

[18] Pachet F., P. Roy, D. Cazaly, "A Combinatorial approach to content-based music selection", Proc. of IEEE International Conference on Multimedia Computing and Systems, Firenze, Italy, Vol. 1 pp. 457-462, 1999.

[19] Pachet, F., G. Westerman, and D. Laigre, "Musical Data Mining for Electronic Music Distribution", WedelMusic Conference, Firenz, Italy,, Nov. 2001.

[20] Pachet F., "Electronic Music Distribution: The Real Issues", Communications of the ACM, to appear, 2002.

[21] Peeters G., S. McAdams, P. Herrera, "Instrument sound description in the context of MPEG–7", Proc. of the International Computer Music Conference, ICMA, 2000.

[22] Peeters G., P. Tisserand, "Selecting signal features for instrument sound classification", Proc. of the International Computer Music Conference ICMA, 2002 (Submitted).

[23] Peeters G., A. La Burthe, X. Rodet, "Toward Automatic Music Audio Summary Generation from Signal Analysis", Proceedings of the International Conference on Music Information Retrieval, Ircam, Paris, 2002.

[24] Tzanetakis, G. "Manipulation, Analysis and Retrieval Systems for Audio Signals", Ph. D. dissertation, Princeton University, 2002.

[25] Wöhrmann, R., G. Ballet, "Design and architecture of distributed sound processing and database systems for web-based computer music applications", Computer Music Journal, vol 23, Number 3, p.77-84, 1999.

[26] Zils, A., F. Pachet, "Musical Mosaicing" Proceedings of DAFX 01, Limerick (Ireland), 2001.