A MULTIMEDIA SEARCH AND NAVIGATION PROTOTYPE, INCLUDING MUSIC AND VIDEO-CLIPS

G. Peeters, F. Cornu
Ch. Charbuillet, D. Tardieu, J.J. Burred
STMS IRCAM-CNRS-UPMC
{peeters,cornu}@ircam.fr

M. Vian[1], V. Botherel[2]
J.-B. Rault[2] and J.-Ph. Cabanal[2]
[1]Bertin Technologies, [2]Orange-Labs
jeanphilippe.cabanal@orange.com

ABSTRACT

Moving music indexing technologies developed in a research lab to their integration and use in the context of a third-party search and navigation engine that indexes music files, archives of TV music programs and video-clips, involves a set of choices and works that we relate here. First one has to choose technologies that perform well, which are scalable (in terms of computation time of extraction and item comparison for search-by-similarity), and which are not sensitive to media quality (being able to process equally music files or audio tracks from video archives). These technologies must be applied to estimate tags chosen to be understandable and useful for users (the specific genre and mood tags or other content-descriptions). For training the related technologies, relevant and reliable annotated corpus must be created. For using them, relevant user-scenarios must be created and friendly Graphical User-Interface designed. In this paper, we share the experience we had in a recent project on integrating six state-of-the-art music-indexing technologies in a multimedia search and navigation prototype.

1. INTRODUCTION

The objective of the MSSE project (Multimedia Search Services for European Portals) is to develop a multimedia search and navigation prototype, which gives access to several types of contents (catch-up TV, archives, videos, music) and which illustrates the benefits of advanced audio-video analysis technologies. The prototype is organized around three use-cases:

- Searching recently broadcasted TV programs ("Catch-up TV"); navigating inside the videos by chapters or keywords.
- Searching video extracts on a specific topic related to recent news and culture; browsing in relevant translated foreign videos and in public TV archives.
- Searching and exploring music pieces with the help of tags, music structure, summaries and similarity.

The prototype is based on video indexing, speech recognition and music indexing technologies. In this paper, we describe the works performed for the music indexing technologies. Those have to deal with three types of content:

- A music collection
- A collection of video clips from the W9 TV channel
- A collection of video archives of music programs from the INA [1] collection. (only the audio track of the video is processed by our indexing modules).

In this paper, we propose to review the technologies integrated into this search and navigation prototype, why they were chosen and how they were developed and integrated as well as the corresponding user-evaluations and GUI developed. We believe that sharing the experience of this work could provide a good example of integration of research modules in a real application scenario.

While many papers have been published on the independent elements this paper deals with (content-based, semantic tags, corpus creation, GUI, user-tests), few of them deal with all these elements as a whole to create a system. Among exceptions are the works made for the Music Browser [1], FM4-Soundpark [2], Musicream [3], MusicBox [4] or PlaySOM [5]. Our work differs from the previous in the number of integrated technologies, the integration into a whole video and music search engine accessible through a web-browser and the simplicity of the GUI.

2. OVERALL DESIGN PROCESS

Figure 1 represents the various elements of work (and interaction/dependency between them), needed for integrating the music technologies in the prototype.

The starting point is a set of requirements from the third-party developer and its users [2] −in terms of functionalities (such as searching-by/ filtering-by tags, search-by-similarity or summarization) and −in terms of types of content-description (genre, mood, instrumentation).

From this, a set of potential technologies are studied in terms of performances and scalability [3] . Candidate technologies are tested over the years in internal benchmarking or in public ones such as MIREX. For example, from

[1] French National Audio Visual Archiving Institution

[2] During the project, 1 or 2 user tests per year were performed, each corresponding to a new version of the prototype (see part 6).

[3] By scalability we mean computation time of content-extraction and of items comparison for search-by-similarity.

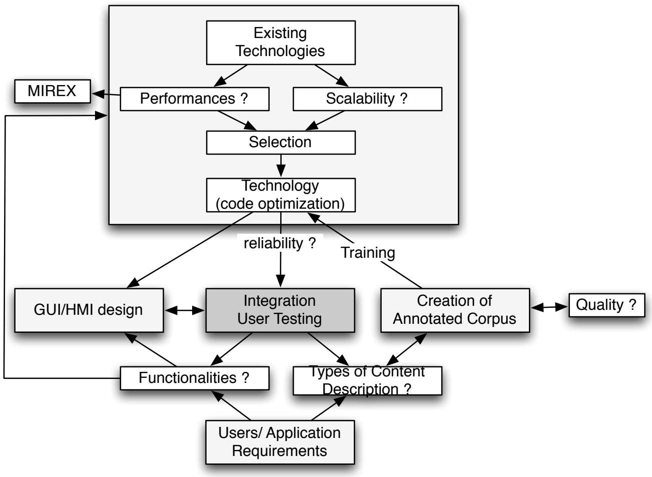


Figure 1. Interaction/dependency between the various elements of work needed integrating music technologies.

our tests in MIREX between 2008-2011, it appears that using Universal Background Model (UBM) to model audio features [6] has many advantages over other techniques: it achieves performances among the best for both auto-tagging [7] and similarity tasks [8]; it allows to share the same front-end for both tasks; it allows easy scalability in the case of similarity (items comparison remains in an Euclidean space). Therefore, we chose UBM for these tasks.

In parallel, the design of the GUI starts. Since the GUI directly infers on the usability of the functionalities, its design is mainly driven by those. It is also driven by the outputs of user-tests and by extra outputs that technologies can provide without extra-costs. For example, when computing audio summaries, music structures are estimated as an internal step. Therefore, it can easily be integrated to provide new functionalities (display interactive player).

In a latter stage, annotated corpora need to be created for each of the requested content description (genre, mood, instrumentation). This part forms a close feedback loop between: – annotation of a corpus, – measuring the reliability of the annotations (this can highlight the fact that some required concepts may appear unclear), – redefining the types of content with the third-party. After several iterations, this loop-process leads to a much clearer set of content-description concepts (the specific definition of genre, mood, instrumentation) and more accurate annotated corpora (their specific use for music tracks).

These annotated corpora are then used to the train the corresponding technologies and optimization is performed to reduce computation time, disk access and memory load.

The resulting prototype is then submitted to global user tests (testing both functionalities, the GUI and the underlying technologies to achieve the functionalities). The whole process is then started again (once a year in our project).

3. TECHNOLOGIES INTEGRATED

Resulting from the process explained in part 2, six different music-content-based technologies have been selected:

- auto-tagging based on training (for genre, mood, instrumentation tags and singing segmentation),
- two technologies for auto-tagging based on dedicated models (for tempo and key/mode tags),
- search by similarity (for music recommendation),
- music structure (for interactive browsing),
- audio summary creation (for content preview).

These modules are either applied to mp3 files or to the audio part of video archives or clips. The inter-connections between the various modules are indicated in Figure 2. It should be noted that the first five technologies were evaluated very positively in the recent MIREX-11 evaluations.

3.1 Audio feature extraction

In order to decrease the total computation time, auto-tagging based on training and search-by-similarity are based on the same audio features front-end. The audio features front-end is described in Figure 2 and corresponds to the proposals made in [8], [7] or [9]. It is based on two modeling techniques coming from speech processing:

• **Universal Background Model (UBM)** [6] [10]. The aim of this technique is to represent the "world" of features using a GMM and then deform[4] this "world" to represent a new feature vector. The resulting representation is the concatenation of the adapted μ-vectors of the GMM, the size of which depends on the dimensionality D of the initial feature vectors and the number m of mixtures used for the GMM. These concatenated-vectors are denoted by "Super-Vectors" (SV) in the following.

• **Multivariate Auto-Regressive Model (MAR)** [11]. As for the mono-dimensional AR-model, the goal is to represent the dependency of the values of a signal over time by an all-pole filter of order K[5]. In the case of the MAR, we consider the dependencies in time and between the various D dimensions of the feature vectors. The results of this is a matrix of coefficients $\underline{\underline{\alpha}}_{k,d}$.

The input to these two modeling techniques is a feature set made of 13 Mel Frequency Cepstral Coefficients and 4 Spectral Flatness Measure coefficients, extracted using a 40 ms Blackman window with a 20ms hop size. From those, two modeled feature sets are computed: (1) Super-Vector of MFCC/SFM, which we denote by SV(mfcc/sfm), (2) MAR of MFCC/SFM, which we denote by MAR(mfcc/sfm). The two modelings are performed using – either the whole set of features inside a track (in case of search-by-similarity and global auto-tagging) – or the set of feature inside successive windows of 2s duration (in case of segmentation, such as singing voice location). In each case, the UBM has been previously trained on a representative database. This training is the most time-consuming part but needs only to be performed once. The UBM configuration is a set of $m = 64$ (for search-by-similarity) or $m = 32$ (for auto-tagging) mixtures, each with a diagonal covariance matrix. The order of the MAR model is $K = 4$.

[4] Deforming means here adapting the μ-vectors of the GMM using an Expectation Maximization algorithm.

[5] $s(n) = \sum_{k=1}^{K} \alpha_k s(n-k) + \epsilon$ where s is a signal, n discrete time, ϵ a residual.

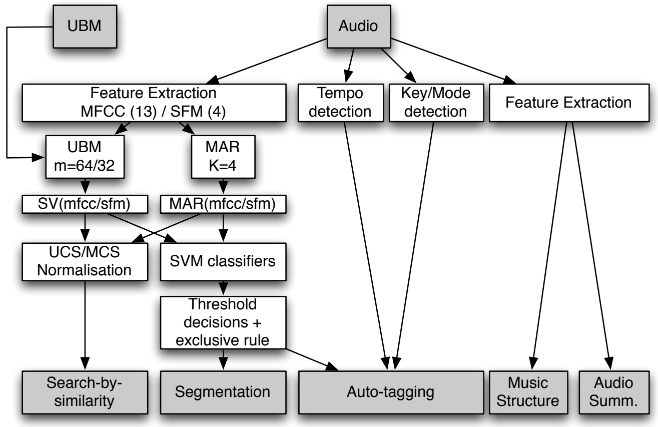


Figure 2. Modules used for music content extraction.

Auto-tagging based on training		
Categories	**Configuration**	**Tags**
Genre	single-label	Classical, Other Genres, NA
Other Genres	multi label	Pop/Rock, Blues, Electronica, Metal/Punk, Reggae, Jazz, Rap, Soul/Funk, Rhythm & Blues, Latin/Bossa
Mood	single label single label single label	Happy, Sad, NA Dynamic, Calm, NA Romantic, NA
Instrumentation	multi label	Brass, String, Piano, Electronic, Acoustic
Drum Kit	single label	No drum, Electronic, Pop/Rock, Hard/Metal
Guitar	single label	No guitar, Acoustic Guitar, Electric Guitar
Live Studio	segmentation + single label	Live, Studio
Singing	segmentation	Singing voice

Table 1. Categories, configurations and tags of the various classifiers used for the Auto-tagging modules

3.2 Search by similarity

As explained in [8], the main goal of using UBM and MAR for modeling the features (instead of the usual MFCC/GMM with EMD Kullback-Leibler divergence) is to remain in an Euclidean space. In the case of search by similarity, it therefore allows the use of standard techniques to decrease the search time in the database. In order to avoid hubs and orphans, various techniques have also been proposed. We have used the UCS-norm (UBM Centered Spherical normalization) and the MCS-norm (Mean Centered Spherical) proposed in [8]. Both techniques consist in projecting the features vector on a unit sphere (either centered on the mean of the UBM, or the mean of the database). After this, each track of the database sees the rest of the database with the same point of view (unit sphere). Using those, the similarity between two tracks is simply the correlation of their vectors. Two similarity matrices, corresponding to the two feature sets are then computed and combined linearly (late-fusion).

3.3 Auto-tagging based on training

Auto-tagging based on training aims at providing the tags indicated in Table 1. Tags can be exclusive (such as "dynamic" and "calm") or inclusive ("pop/rock" and "electronica"). In our system, all problems are solved using multi-label classifiers in a one-against-all strategy (true versus false class). For this, all problems are decomposed as set of binary SVM classifiers (with an RBF-kernel, σ=1) [12] [7]. The input to the classifiers is the concatenation of SV(mfcc/sfm) and MAR(mfcc/sfm) (early fusion).

Global Classifiers: Music **genre** classifier is a set of 11 binary classifiers (one for each genre) trained and evaluated independently. A given track t is said to belong to a tag-class c if the affinity-output $a_c(t)$ of the corresponding SVM classifier is above a threshold A_c. The estimation of each threshold A_c is based on the Recall/Precision curve obtained on a training set. Considering that the estimated tags are to be used as search criteria, it was decided to favor Precision over Recall: we chose the lowest A_c leading to a Precision greater than 0.8. In terms of usability, we also decided to make "classical music" mutually exclusive to the "other genres" (see Table 1). For a given track t, if both $a_{class}(t)$ and several $a_{other}(t)$ are above their respective threshold, the choice is based on the maximum between $a_{class}(t)$ and $\max(a_{other}(t))$. In case $\max(a_{other}(t))$ is selected, the corresponding subgenres above their respective thresholds are returned. The same process is applied for the 5 **mood** classifiers. In this case, the mutually exclusive classes are "happy" / "sad" and "calm" / "dynamic". The auto-tagging module also returns three view-points related to the **instrumentation** of the track: (1) a global instrumentation based on dominant instruments (brass, string, piano, electronic instruments, acoustic instruments), (2) a detailed description of the percussive part (electronic drum, pop/rock drum, hard/metal drum) (3) a detailed description of the guitar part (acoustic guitar, electric guitar).

Segmentation: The segmentation is obtained by detecting class-changes over time. For this, the same system as presented above is used, but the UBM/MAR models are applied to the set of features inside a succession of windows of 2s duration (hop size of 1s). Each 2s features is then classified using SVM classifiers. This segmentation is used to provide **singing/**non-singing segmentation over time. In order to avoid spurious class transitions over time, a 3rd-order median filter is applied to the estimated classes over time before segmentation. This segmentation is to be used to display singing segments in the interface.
We also use this segmentation to perform the **"live/studio"** auto-tagging. In our case, "live" is defined as the presence of "applauses, whistling ..." of audience in a bar, concert-hall, stadium. Since those do not occur over the whole time-duration of the track (usually at the beginning, ending or during a break), the decision is based on frame-classification. We use a minimum threshold of 26s frames being classified as "live" for the track to be classified as "live". A similar approach has been used in [13].

Each tag has also an associated "reliability" defined in the interval $[0, 1]$ (low/high reliability). For this, the affinity of each SVM is passed through a sigmoid and centered on its respective threshold. This reliability is to be used by the GUI for sorting the list of results.

3.4 Auto-tagging based on dedicated algorithms

For each track, we also estimate its global **tempo** in beats-per-minute. Note that this estimation does not rely on the set of UBM/MAR features but on a dedicated algorithm. We have used the algorithm proposed in [14]. We also assign a "reliability" to this estimated tempo. For this we used the "periodicity" features proposed by [15] (measurements of the amount of periodicity in the track).

We also estimate the global (most dominant over time) **key/mode** among a set of 24 key/mode classes (C Maj, C min . . . B Maj, B min). We have used the algorithm proposed in [16]. The "reliability" of the output is here estimated as the distance between the most-likely key/mode and the second most likely.

3.5 Music Structure and Summary

This module aims at providing two functionalities: (1) to display a map of the temporal organization of the track (music structure) which allows user to interact with it (skip forward/backward by parts) [17], (2) to provide a meaningful preview of the track content (music audio summary). The estimation of the music structure and of the audio summary are based on the same front-end. This front-end combines the three similarity matrices corresponding to MFCC, Spectral-Contrast and Spectral-Valley [18] measures and Chroma/Pitch-Class-Profile (see [19] for details).

Music Structure Estimation: For robustness reasons, the structure is estimated using a "state" approach. For this, a segmentation of the similarity matrix is first performed using a "checker-board" kernel [20]. The segments obtained are then grouped using a constrained hierarchical agglomerative clustering. The distance used for this clustering is a linear combination of − the distance between the average values inside the two segments (centroid linkage) − the smallest possible distance between one of the diagonals they may contain (sequence approach) − a constraint to minimize the departure of the duration of the merged segments from the average segment durations.

Music Audio Summary Generation: The technique used for the summary creation is based on an extension of the summary score of [21]. In this extension, the method of [21] is iteratively applied to the combined matrix of [19]. At each iteration, the two time corridors in the self-similarity-matrix corresponding to the previously chosen audio extract are canceled to avoid further re-uses. To generate the final audio signal, the selected segments are concatenated using a Downbeat Synchronous OverLap-Add (DSOLA) techniques.

4. ANNOTATED CORPORA FOR TRAINING

4.1 Corpus creation for the UBM training

Since both auto-tagging and search-by-similarity modules rely on Super-Vectors, the corresponding UBM needs to be trained in advance. The training of which needs to take into account the various types of contents (various genres and various audio qualities) that the system will need to deal with. For this, a large database of audio files has been used: including clean mp3 files at various bit-rates and audio tracks of TV archives.

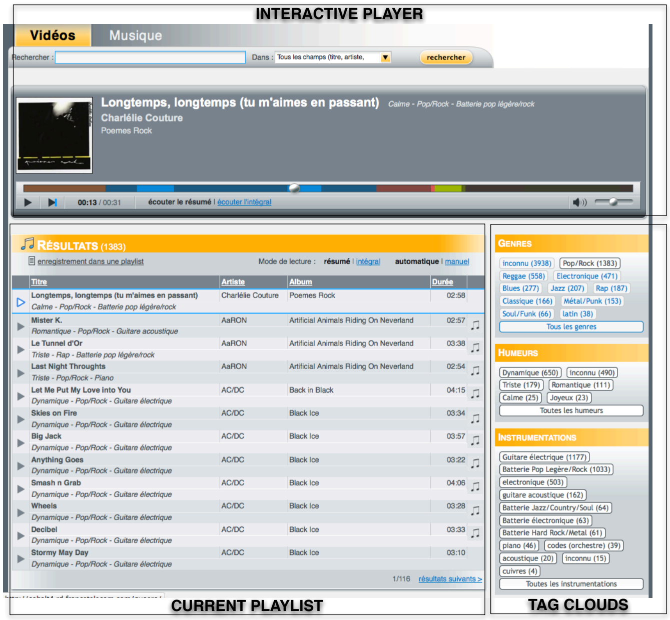
4.2 Annotated corpora for the auto-tagging problems

For the auto-tagging modules, statistical models (SVMs and related thresholds) need to be trained for each tag (genre, mood, instrumentation, singing, live). We explain here the data used for the training. For the creation of the list of **genres**, several attempts have been made: − from a purely acoustic definition of genres (pop-rock synthed, poprock hard, electronica ambient, electronica beat . . .) which guarantees a close proximity to content-based estimation algorithms but may be difficult to understand by users − to a purely application oriented definition. The final list indicated in Table 1 is the results of a feedback-loop between the two. The training-set has then been obtained by selecting tracks among a large music collection considered as prototypical of the chosen genres. By prototypical, we mean tracks representative of the exact genre and not cross-over between several genres. For the other tags (**mood, instrumentation**), 4000 tracks have been manually annotated by two individual professional annotators. Only labels for which the annotators agreed on the majority of the tracks are considered. For these labels, only tracks for which both annotators agreed have been selected for the training. This process lead to the five moods and three view-points on instrumentation indicated in Table 1. "**Live**" classifier has been trained on a dedicated training-set made of the concatenation of all possible audience noise derived from real recording. The **singing** segment classifier has been trained using the Jamendo corpus [22].

5. GRAPHICAL USER INTERFACE

The GUI is the central element that allows user to interact with the prototype and to test the proposed use-cases. Its design is crucial since a bad GUI can hinder a good technology or a good use-case. Its design must follow a close user-feedback loop (see part 6). The current GUI (see Figure 3) is organized in three main panels: the interactive-player (top), the current play-list (left), the various tag-clouds (right).

The **player** panel displays the classical editorial metadata (track-title, artist-name, album-title) and the cover. A large horizontal time-line displays the estimated structure of the tracks. In this, parts with similar content are indicated by rectangle with similar colors. The user can browse through parts by directly clicking on the corresponding colored rectangle. The time-line also indicates the segments used for the audio summary by highlighting the corresponding parts (independently of the color). Once selected (using the play-list panel), a track automatically starts playing in the player either in full-duration or in audio summary mode. This choice is based on user preferences. A search-by-text panel is placed on the top of the


INTERACTIVE PLAYER
Vidéos
Musique
Rechercher :
Dans : Tous les champs (titre, artiste,
rechercher
Longtemps, longtemps (tu m'aimes en passant) Calme - Pop/Rock - Batterie pop légère/rock
Charlélie Couture
Poemes Rock
00:13 / 00:31
écouter le résumé | écouter l'intégral
CURRENT PLAYLIST
TAG CLOUDS

RÉSULTATS (1383)

enregistrement dans une playlist Mode de lecture : résumé | intégral automatique | manuel

Titre	Artiste	Album	Durée
Longtemps, longtemps (tu m'aimes en passant) *Calme - Pop/Rock - Batterie pop légère/rock*	Charlélie Couture	Poemes Rock	02:58
Mister K. *Romantique - Pop/Rock - Guitare acoustique*	AaRON	Artificial Animals Riding On Neverland	02:57
Le Tunnel d'Or *Triste - Rap - Batterie pop légère/rock*	AaRON	Artificial Animals Riding On Neverland	03:38
Last Night Throughts *Triste - Pop/Rock - Piano*	AaRON	Artificial Animals Riding On Neverland	02:54
Let Me Put My Love into You *Dynamique - Pop/Rock - Guitare électrique*	AC/DC	Back in Black	04:15
Skies on Fire *Dynamique - Pop/Rock - Guitare électrique*	AC/DC	Black Ice	03:34
Big Jack *Dynamique - Pop/Rock - Guitare électrique*	AC/DC	Black Ice	03:57
Anything Goes *Dynamique - Pop/Rock - Guitare électrique*	AC/DC	Black Ice	03:22
Smash n Grab *Dynamique - Pop/Rock - Guitare électrique*	AC/DC	Black Ice	04:06
Wheels *Dynamique - Pop/Rock - Guitare électrique*	AC/DC	Black Ice	03:28
Decibel *Dynamique - Pop/Rock - Guitare électrique*	AC/DC	Black Ice	03:33
Stormy May Day *Dynamique - Pop/Rock - Guitare électrique*	AC/DC	Black Ice	03:10

1/116 résultats suivants >

GENRES

inconnu (3938) Pop/Rock (1383)
Reggae (558) Electronique (471)
Blues (277) Jazz (207) Rap (187)
Classique (166) Métal/Punk (153)
Soul/Funk (66) latin (38)
Tous les genres

HUMEURS

Dynamique (650) inconnu (490)
Triste (179) Romantique (111)
Calme (25) Joyeux (23)
Toutes les humeurs

INSTRUMENTATIONS

Guitare électrique (1177)
Batterie Pop Legère/Rock (1033)
electronique (503)
guitare acoustique (162)
Batterie Jazz/Country/Soul (64)
Batterie électronique (63)
Batterie Hard Rock/Metal (61)
piano (46) codes (orchestre) (39)
acoustique (20) inconnu (15)
cuivres (4)
Toutes les instrumentations

Figure 3. GUI of the Music Interface of the prototype

interface. It allows either full-database search or search over restricted criteria (title, artist, album).

The **play-list** panel indicates the currently selected tracks which correspond either to – the results of a search-by-text, a search/filtering using the tag-clouds or a search-by-similarity, – or a previously stored play-list [6] . For each track, the estimated tags (genre, mood, instrumentation) are also indicated. The musical note icon next to each track allows performing search-by-similarity.

The **tag clouds** panel indicates the various viewpoints on the content: genre, mood, and instrumentation. The tags that are currently active in the filtering are indicated by highlighted colors. Next to each tag-name is indicated the number of corresponding items. It should be noted that the tags inside a cloud are not mutually exclusive.

6. USER TESTS

We define user experience as "the combination between the quality of the technology, the functionality based on it and the way to present it on Human Machine Interface". During the project, 1 or 2 user tests per year were performed, each corresponding to a new version of the prototype.

Many of the outputs of user-tests relate to the usability of the GUI: naming of the fields, their spatial organization, layout of the tag-clouds . . . This is of course essential; especially considering that the music search part is only one part of the whole search engine (which also includes TV and Web-Video search) and the presentation of the various search engines must be as much as possible homogeneous. User-tests are performed using two methods.

6.1 Qualitative tests

The first method consists in performing **qualitative** tests. Qualitative tests have three focuses: (1) to asses the usability of HMI (2) to asses users' judgment of functionalities (3) to imagine with users new use cases and maybe new functionalities based on the music technology. For this, users were asked to perform various scenario: "use the search engine to create a music play list of a specific mood", "to discover new music" . . . This is followed by interviews, which allows highlighting problem in the usability of the GUI, collecting judgments of functionalities (audio summary, genre, mood and similarity are found highly relevant while music structure displaying less relevant). This has also allowed highlighting missing functionalities. Displaying singing segments was one of those.

6.2 Quantitative tests: the case of audio summary

The second method consists in performing **quantitative** tests to compare several variations of a technology. An example of this are the "audio summary" user-tests.

For the creation of the summary, a set of user tests have been performed in order to select the best summary strategy. For this we have compared four different types of summary: –a 30s extract at the beginning of the file, –a random 30s extract, –the most representative 30s extract (denoted by 1x30), –a downbeat-synchronous concatenation of the three most representative 10s extracts (denoted by 3x10) [17]. 24 users had to listen to tracks of music they knew (7 tracks) and music they didn't know (6 tracks). Half of the songs were in their native language (French), the other half in English. They were then asked the questions - "which technique better summarized the track" (for music they knew), - "which technique is the most informative of its content" (for music they didn't know). In both cases, the 3x10 summary was judged better.

A quantitative evaluation has also been performed to compare the 1x30 and 3x10 summary. Over a 160-tracks database, we have measured the number of tracks for which each technique allowed to include the track title in the summary (the track title is considered here as the most memorable part of the track). The 3x10 summary achieved 95% correct location, while the 1x30 achieved 90%.

A user evaluation of the acoustical quality of the multi-parts (3x10) summary has also been performed. We have compared four configurations of the audio construction: –complete DSOLA –partial DSOLA (the loudness of the audio decreases during the transitions between parts to highlight them), –DSOLA with sound insertion (a prototypical sound is introduced at each transition to highlight them), –partial DSOLA with visual feedback. This experiment highlighted the fact that in some cases (especially Rap music), the complete DSOLA leads to an audio that sounds exactly like a real track. However, users prefer to feel a separation between the three 10-second parts to avoid having the feeling of listening to a new mix from a DJ. We also decided to add a visual presentation to increase the understanding of this summary functionality. This visual presentation consists in 3 highlighted segments of the com-

[6] The playlist, tempo and key/mode functionalities are not discussed here since their are currently subject to modifications of the GUI.

plete music timeline corresponding to the three 10-second parts of the summary. The play cursor "jumps" from part to part. With these choices and modifications, user experience of the summary was improved.

7. INTEGRATION

The back-office of the prototype is based on a Service Oriented Architecture (SOA). This kind of architecture is flexible and particularly adapted for the integration of numerous and distant technologies. The main elements of this architecture are: −Metadata collectors, which collect metadata coming from content providers (TV Programs, INA archives, Web videos, music); −Technological modules, accessible as Web services (e.g. speech to text, named entities extraction, music analysis); −An XML transverse metadata base, which stores all metadata coming from collectors and technological modules; −An ESB (Enterprise Service Bus), which connects the metadata collectors, the technological modules and the metadata base; −A specific XML "pivot" format for all metadata manipulated by the ESB and the XML database. The search engine indexes are fed by the XML database through a metadata exporter. The search engine is directly connected to the application.

8. CONCLUSION

In this paper, we wanted to share our experience on integrating music-content indexing technologies, as developed in a research lab, into a third-part search and navigation engine. For this, we provided a panorama of the various elements of works implied and how they interact.

The lessons we learned from this experience is that this integration involves much more than good signal processing and machine learning technologies, which are of course essential. A side from the technical constraints (robustness, scalability), many of the works to be performed relate to make these technologies usable. This involves first proposing useful and understandable tags for users and creating the related annotated corpus to train the algorithms. This also involves tuning and modifying technologies: to favor precision over recall; or to provide reliability for all estimations (which is difficult for descriptions such as tempo or key). User tests allows to highlight new challenges, such as the need for a list containing only the similar items and not just a ranked-list from the most to the less similar items; or the fact that some innovative technologies may be found too specialized for users (music structure). We hope the information provided here would help the research community when trying to move from research applications to third-party applications.

Ackowledgements: J.J. Burred is now with Audionamix. This work was supported by the "Quaero" Program funded by Oseo French State agency for innovation.

9. REFERENCES

[1] F. Pachet, J. Aucouturier, A. La Burthe, A. Zils, and A. Beurive, "The cuidado music browser: an end-to-end electronic music distribution system," *Multimedia Tools and Applications*, vol. 30, no. 3, pp. 331–349, 2006.

[2] M. Gasser and A. Flexer, "Fm4 soundpark: Audio-based music recommendation in everyday use," in *Proc. of the 6th Sound and Music Computing Conference (SMC 2009), Porto, Portugal*, 2009.

[3] M. Goto and T. Goto, "Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces," in *Proceedings of the 6th International Conference on Music Information Retrieval*, pp. 404–411, 2005.

[4] A. Lillie, *MusicBox: Navigating the space of your music*. PhD thesis, Massachusetts Institute of Technology, 2007.

[5] P. Knees, M. Schedl, T. Pohle, and G. Widmer, "Exploring music collections in virtual landscapes," *Multimedia, IEEE*, vol. 14, no. 3, pp. 46–54, 2007.

[6] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[7] D. Tardieu, C. Charbuillet, F. Cornu, and G. Peeters, "Mirex-2011 single-label and multi-label classification tasks: Ircamclassification2011 submission," in *MIREX Extended Abstract*, (Miami, USA), 2011.

[8] C. Charbuillet, D. Tardieu, and G. Peeters, "Gmm supervector for content based music similarity," in *Proc. of DAFX*, (Paris, France), pp. 425–428, September 2011.

[9] C. Charbuillet, D. Tardieu, F. Cornu, and G. Peeters, "2011 ircam audio music similarity system 1," in *MIREX Extended Abstract*, (Miami, USA), October 2011.

[10] C. Cao and M. Li, "Thinkit's submissions for mirex2009 audio music classification and similarity tasks," in *MIREX Extended Abstract*, (Kobe, Japan), 2009.

[11] F. Bimbot, L. Mathan, A. De Lima, and G. Chollet, "Standard and target driven ar-vector models for speech analysis and speaker recognition," in *Proc. of IEEE ICASSP*, vol. 2, pp. 5–8, 1992.

[12] J.-J. Burred and G. Peeters, "An adaptive system for music classification and tagging," in *Proc. of LSAS (Int. Workshop on Learning the Semantics of Audio Signals)*, (Graz, Austria), 2009.

[13] F. Fuhrmann and P. Herrera, "Quantifying the relevance of locally extracted information for musical instrument recognition from entire pieces of music," in *Proc. of ISMIR*, (Miami, USA), 2011.

[14] G. Peeters, "Template-based estimation of time-varying tempo," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 158–158, 2007. doi:10.1155/2007/67215.

[15] G. Peeters, "Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 19, pp. 1242–1252, July 2011.

[16] G. Peeters, "Chroma-based estimation of musical key from audio-signal analysis," in *Proc. of ISMIR*, (Victoria, Canada), pp. 115–120, 2006.

[17] G. Peeters, A. Laburthe, and X. Rodet, "Toward automatic music audio summary generation from signal analysis," in *Proc. of ISMIR*, (Paris, France), pp. 94–100, 2002.

[18] D. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast," in *Proc. of IEEE ICME (International Conference on Multimedia and Expo)*, (Lausanne Switzerland), 2002.

[19] G. Peeters, "Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach," in *Proc. of ISMIR*, (Vienna, Austria), 2007.

[20] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. of IEEE ICME (International Conference on Multimedia and Expo)*, (New York City, NY, USA), pp. 452–455, 2000.

[21] M. Cooper and J. Foote, "Automatic music summarization via similarity analysis," in *Proc. of ISMIR*, (Paris, France), pp. 81–85, 2002.

[22] M. Ramona, G. Richard, and B. David, "Vocal detection in music with support vector machines," in *Proc. of IEEE ICASSP*, pp. 1885–1888, 2008.