Database-Driven Systems and Applications for Music Information Management and Retrieval

by

Otto Wüst

Submitted in partial fulfillment of the requirements for
the degree of Diploma of Advanced Studies
Doctorate in Computer Science and Digital Communication
Department of Technology

Advisor: Dr. Xavier Serra

Universitat Pompeu Fabra

Barcelona, September 2003

Abstract

Many scientific and technical developments in recent years have lead to a situation where moderately large collections of multimedia data, in particular of musical data, are available to computer users, who start to demand new applications that can go beyond the traditional file system based data management. This work analyzes the problems and scientific contributions in the field of music information retrieval, specifically focusing on database related issues and applications, to establish the state of the art. Furthermore, and due to the character of research retrospective, this document reviews some of the author's work in the area of music and database applications, including the implementation of an MPEG-7 database layer, a crawler system which gathers musical information of web pages that is then used to construct a cultural similarity measure, and a database driven architecture for collaborative collective composition on the web. Finally, conclusions are drawn and some research problems that constitute the author's interest for future work are exposed.

Acknowledgements

Conducting research is a gratifying task. Frequently, however, it is not easy to find either the time or alternatively the resources that will grant you the time for doing it. This is specially true for somebody working full time in a consulting division of a large international corporation.

I would like to thank all the people that have contributed to this work, either by sharing research and work experience on the subject matter, either by granting time and resources for me to spend on the work, or by speaking the adequate support words at the right moment.

Very special thanks go to my family and friends, my colleagues at Oracle and at the MTG, the reviewers of this report, and my advisor Dr. Xavier Serra.

Otto Wüst
Barcelona, September 2003

Table of Content

Table of Figures

Chapter 1 Introduction

Content based retrieval of multimedia data, besides being a popular and attractive research topic, is rapidly becoming a real need for many computer users. In the particular case of audio and music, not only professional studio owners or electronic music distribution companies face themselves with large collections of music files, but also home users seem to be accumulating large amounts of musical data. Some factors that contribute to this trend could be the low cost of storage space, the popularity of the mp3 file format and proliferation of mp3-based portable devices and, despite the shutdown of Napster and Audiogalaxy, the non-negligible traffic of file exchanges through p2p networks.

The main issue from a functional point of view is that it is not practical to manage large collections of multimedia data, audio in particular, with the standard file system based tools. When searching for a particular file, most users have to issue the search based on the filename, which may easily not be representative of the file content, even if it wasn't set arbitrarily at origin. Searches on a filename are typically implemented with exact string matching algorithms, and the most advance feature is the use of wildcards. Let's illustrate just a fraction of the common problems with a simple example in which a user is trying to retrieve some music titles by the jazz musician Charlie Parker:

- **lack of selectivity**: the query string "*Charlie*Parker*" will likely retrieve an excess of titles
- **sensitivity to misspelling**: nothing will be possibly returned by query string "*Charly*Parker*" due to an error in the name that a non expert user may not be aware of
- **errors in the manual annotation**: the query expression may be correct, but potentially the desired file was named incorrectly due to an error in the manual or automatic processing when the file was "prepared" and/or stored
- **order of characters in query expression**: the query strings "*Charlie*Parker*" and "*Parker*Charlie*" are not equivalent
- **semantics of the topic**: given a query using Parker's nickname "*Bird*" it may fail to retrieve results if no file contains the nickname's characters
- **semantics of the user**: a query or a file name could express the context in which that particular song has a meaning for the user "Christmas party 2001" while it is likely that no files exist with those names

And this list of problems doesn't even include any of the reasonable content based retrieval requirements a user could have. The most recent versions of common operating systems and specialized library management software are introducing the possibility to use some of the tags contained in mp3 files, and this definitely improves the quality of file system or (more precisely) simple text label based retrieval. However, potentially the demands of users are clearly beyond what is currently commercially, or even technically and scientifically available. In a typical music listening situation, one may want to create a play list where all the songs have something in common, the same beat, a similar instrumentation, the same musical style, same energy, all sharing an ABABCA structure, all featuring a mandolin solo, etc.

In the previous examples we are interested in retrieving some songs by structural, cultural, or perceptual criteria of similarity. At this point, it is difficult to imagine that mp3 files (even in some future version of the format) will contain these type of annotations. It may even be questionable if some of these annotations make much sense at all, as some of them may not even be objective, and could therefore be quite variable, for example depending on the experiences of the user who labeled them. There appears to be much room for improvement. The MPEG-7 standard has been recently introduced to provide an interface for describing the content, however it does not say how to implement the systems that must support it and many aspects of how to do that still remain unclear. Currently only prototype implementations exist, one of which is described later in this work.

The field of music information retrieval research generically deals with all these kind of questions and issues. This research work, while falling in the MIR research category, has a particular focus on some elements and aspects which are of specific interest to the author: database driven systems and applications for retrieval and management of the musical information.

The main objective of this research report is to review some of the already known problems and approaches in the subject of interest, in order to determine the state of the art. This should help to better understand where new contributions should be made and therefore serve as a basis for future research work. Chapter 2 reviews the subject area and approaches of classic information retrieval, and then various representative applications of music information retrieval, including identification, search by melody, browsing of sounds and visualization. A review of the MPEG-7 standard for describing musical content closes the chapter.

A further objective, covered by Chapter 3, is to present some of the contributions that the author has made during the last four years and which correspond to technical and scientific results obtained from the author's participation in several research projects within the subject area of interest, which may also serve as a starting point for a future doctorate work.

Finally, some conclusions are drawn in Chapter 4 and open issues as well as further ideas for future work are exposed.

Before finishing this section, it is necessary to note that this whole research report is a contribution in partial fulfillment of the requirements for the degree of Diploma of Advanced Studies within the Doctorate in Computer Science and Digital Communications program at the Universitat Pompeu Fabra.

The author

The author of this research work, Otto Wüst, has a degree in Telecommunications Engineering from the Universitat Politècnica de Catalunya. After a year as a research associate at Duke University, he joined Oracle in 1997 where he has held a full time position since, currently as a Principal Consultant specialized in database technology and applications. Since 2000 he also teaches the Data Structures course in the studies of Computer Science as an associate professor at Universitat Pompeu Fabra.

His interest in performing research in technology and being a hobby musician, producer and recording engineer may explain his motivation in joining the Music Technology Group at Universitat Pompeu Fabra in 1999 and enrolling the Doctorate program. This brief bio may as well justify his particular research topics, directly related to music issues and applications, but always centered around database technologies which constitutes his current profession.

Chapter 2 State of the art

2.1 Introduction, scope and objectives of this report

Music information retrieval (MIR) is a broad multidisciplinary research field that serves as an umbrella for many research activities in distinct areas that range from musicology, through signal processing and communications applied to music, to diverse fields within computer science and library sciences. All of the MIR research however has one common interest: MIR deals with the problems related to appropriately representing and manipulating audio data for its adequate storage and retrieval in computer systems.

Of particular interest from a functional point of view is the notion of "content based retrieval", which comprises the issues related to retrieving rich multimedia information not by labels such as the filename, but instead by using features of the actual content. In the case of MIR this could mean to retrieve only those pieces in a collection that contain a particular melody or a given variation of that melody, or to retrieve fragments in which a given instrument plays a certain note. These first examples are in fact quite objective and we could think of a system that could somehow annotate all this information while it is being generated at the source. A further level of complexity is introduced when perceptual features such as the timbre come into play, as in this case a retrieved content may or not be a correct target depending on the user's training or prior experience.

This report takes into account the author's research interests and presents a review of Information Retrieval and in particular some selected Music Information Retrieval applications, which at this point should represent the state of the art in those areas.

The primary objective is that the present review work can be further used to detect areas where there is still space for proposing new research contributions.

Due to the multidisciplinary nature of MIR, several overlaps have emerged in the process of creating proper taxonomies of the field, and it has not been always easy to maintain a clear overview on all the existing contributions to this area of research. The MindManager© mind mapping tool has been used to assist the analysis process and structuring of this report into topics and subtopics and finally the actual contributions, some of which are overlapping, and some of which cover multiple topics.

This chapter is organized as follows. First a review is made on the concepts of classical (mostly text based) information retrieval and its evaluation. A discussion of overview works on music information retrieval follows. Finally, several typical applications in the field of MIR are discussed in detail, including Identification, Search by Melodic Similarity, Browsing in Timbre Space, Visualization and Recommendation. Finally the MPEG 7 Standard is reviewed as a means to provide a framework for deploying MIR applications.

As in the remaining work, in this chapter the primary interest is on database issues, rather than on algorithms. With regards to the content, the discussion is generally

restricted to either polyphonic music titles, monophonic sounds or short sequences or simple rhythm loops.
Finally, it is important to note that this state of the art does not review specifically common database principles or technology, as this has been covered exhaustively in diverse textbooks, such as [Ullman 1982], [Date 1994] or [Subrahmanian 1999].

2.2 Review of classical information retrieval and its evaluation

Extensive work exists in the field of traditional information retrieval. Classic references are [van Rijsbergen 1979] and [Salton 1983]. A good overview on modern information retrieval is provided by [Baeza 1999], while partial aspects involved in the process of information retrieval such as data structures and algorithms, indexing and compression, and digital libraries and its applications as well as its economical impact are respectively discussed in [Frakes 1992], [Witten 1994] and [Lesk 1997].

2.2.1 Information retrieval models

An information retrieval model is formally typically defined [Baeza 1999] as a set of logical views or representations for the documents in a collection, a set of queries (which constitute a representation of the user's information needs), a framework for modeling document representations, the queries and their relationships, and finally a ranking function which defines an ordering among the documents with regard to a given query.

Many models have been proposed and represent different approaches to information retrieval. The classical models are the Boolean, the vector and the probabilistic models and are discussed in many texts, including [van Rijsbergen 1979] and [Salton 1983]. The Boolean model is built on set theory and Boolean algebra. Binary decisions are made to determine if a document is or not relevant for a query, depending on the existence within the documents of a given index term used on the query. The Boolean model only retrieves exact matches and does not provide an adequate ranking. It is therefore considered rather a data model instead of an information model and not too adequate for information retrieval, yet it is the most widely used in current commercial systems.

The vector model [Salton 1983] was introduced to overcome the limitation of binary decision. In this model, both the documents and the queries are represented as weight vectors of dimension t (where t is the number of index terms in the system). A user query is evaluated by computing a distance of the query vector with all the document vectors in the system and by ranking the result. A typical function used in the distance computation is the cosine of the angle between the vectors. In this way, the vector model allows to return documents that only partially match the query. Several approaches exist for choosing the proper weighting scheme for the terms. Effective results can be obtained by taking into account the number of times a term appears in different documents of the collection, as well as the repetitions of the terms within a single document.

The probabilistic model was anticipated by Maron and Kuhns [Maron and Kuhns 1960], formalized by Robertson and Sparck Jones [Robertson and Sparck Jones 1976] and finalized by van Rijsbergen [van Rijsbergen 1979 – Chapter 6] and is based on the assumption that given a user query there will be a subset of documents within the collection which are relevant to the query. These represent the correct answer to the user's information retrieval needs expressed by the query and are labeled R. The model will estimate the probability that a given document pertains to the set R and will rank the documents in decreasing order of the estimated probability of relevance. An IR system using a probabilistic model will use the index terms in the query to make an initial estimate of the relevant documents which can be presented to the user who will provide feedback on the real relevance of the retrieved content. This can then be used to improve the estimation. A good overview of probabilistic modeling is provided by [Crestani 1998], a survey over 30 years of research in probabilistic models for information retrieval, which concludes that novice users making use of current generation probabilistic IR systems will on average yield better results than expert users of a Boolean system over the same collection.

Additional models have been proposed to improve or overcome some limitations of the three classical information retrieval approaches. In [Ogawa 1991] fuzzy set theory is used in combination with a thesaurus in form of a keyword connection matrix (a term-term correlation matrix) as an alternative to Bayesian set theory to model approximate matches and achieve ranking. In general, each of the index terms can be associated to a fuzzy set with a given degree of membership within that set, which can vary between 0 and 1. This is equivalent to having clusters of terms where the boundaries are not precisely defined, which allows to retrieve a relevant document even if the exact terms used as the query are not present within the document.

The extended Boolean model is introduced by Salton, Fox and Wu [Salton 1983b] and is positioned as an intermediate between the Boolean and the vector model. In the former model, set operations are taken strictly and relevant documents may not be retrieved. For example, in the case of the conjunction *term1 AND term2*, if only one of the terms appears in a document then it will be discarded as not relevant. In the extended Boolean model, similarity computations are used to model the Boolean operations. By representing the documents as vectors and using a parametrized distance function, the behavior of the system can be varied at query time from strictly Boolean-like ranking to a vector-like ranking.

One of the assumptions generally made is that terms used in the mapping of documents are independent with each other, although this does clearly not correspond to what happens in the real world. In [Salton 1982] Salton discusses several models which take into account dependencies within term pairs or triplets. The generalized vector space model [Wong 1985] relaxes the original vector model by considering that the index term vectors are linearly independent but not necessarily pairwise orthogonal as required by the original formulation.

A further alternative in the area of algebraic models is the latent semantic indexing model [Furnas 1998]. It is based on the theory of single value decomposition and proposes a mapping of the document vectors and query vectors represented in the index term space into a smaller space of concept terms. This reduction should serve as a semantic conceptualization of the terms and should allow to retrieve documents that do not contain any of the index terms used in the query.

In the area of probabilistic information retrieval modeling several alternative approaches have been made using network-based retrieval models. Turtle and Croft [Turtle 1989] [Turtle 1991] introduce an inference network which is adapted to support document retrieval and compare it to conventional Boolean and probabilistic models. Two networks are used, a document network and a query network. The former is only built once for a given document collection and represents the collection using a variety of document representation schemes. This network does not change during query processing. On the contrary, a new query network is built to model each new information need by the user. It consists of one single query node that represents the user's information needs, and it can contain one or more query representations modeling the user's information needs. In addition, the query network is modified as the user interacts with the system to refine the results. This network based approach is generalized and extended by Ribeiro-Neto and Muntz in [Ribeiro-Neto 1996] to achieve improved retrieval performance.

Finally, subject specific information retrieval models have been introduced to address the particularities of a specific field or application. Structured text retrieval models take into account the organization of the content within the text document. They allow to restrict searches within specific sections or pages of a document and to take into account the proximity of index terms in the document. Some examples are the model based on non-overlapping list proposed by Burkwski [Burkowski 1992] and the model based on proximal nodes by Baeza-Yates and Navarro[Baeza 1996]. Other models have been also introduced for the specific task of browsing, in order to address the user's interest in exploring and in finding new references. This is particularly interesting in the context of the world wide web, where the hypertext structure can be taken into consideration by the retrieval engine.

In the area of multimedia information retrieval [Meghini 2001] presents a study which aims at promoting the integration of the various methods and techniques for multimedia information retrieval. This particular model specifically targets to achieve retrieval performance over text and image media; no particular mention is made to the matter of audio or music. The author however claims that it should be possible to generalize his model to media such as audio and video. The model itself is formulated in terms of a fuzzy description logic, which allows to model semantic-based retrieval, as especially required by the multimedia data.

2.2.2 Information Retrieval Architectures

When considering to build a multimedia information retrieval system or, in particular, a music information retrieval system, a starting point can be obtained from the study of the architecture of a system for textual information retrieval. Many such systems have been built over the years and numerous reports have been written to describe their architectures, as for example [Hollaar 1976], some of which are nowadays outdated.

According to [Baeza 1999] a typical text information retrieval system can be architected with the following modules:

- A **user interface** is used for imputing the query, for presenting the retrieval results and optionally to generate relevance feedback by the user on the results.

- A **preprocessor module** which allows to parse and transform the user input into text elements that can be directly manipulated by the indexing system. This can further include a syntactical and lexical analysis, elimination of stopwords (frequent words in language such as articles or prepositions, which are commonly contained in many documents, and which are therefore not selective and thus not adequate for indexing), removal of prefixes and suffixes also called stemming, and index term selection in order to provide a logical view of the query.

- In addition, some further operations to improve the quality of the query entered by the user, such as query expansion, usage of a thesaurus to identify the key concept of the word to be indexed, or term reweighting can be performed in a specific **query management module**, although some of these tasks could also have been handled by the preprocessor section. The query management module would also take into account the relevance feedback management.

- Furthermore, a database management system supporting IR should provide an **indexing module**, which in the case of textual information retrieval will typically be an inverted file, as well as searching methods and structures for string and pattern matching.

- A **module for ranking** will host the information retrieval model implementation and will order the result set according to the computed or estimated relevance for the user.

2.2.3 Evaluation

A brief review is made here on the methods and measures that are commonly used to evaluate the performance of information retrieval systems. A focus is set on retrieval performance evaluation, thus common performance considerations, such as the tradeoffs between response time and storage space, which are also present in IR systems, are not further discussed here.

Measures

Two of the most common measures that are used to evaluate the quality or goodness of a retrieval operation are the Recall and Precision. Given a collection of documents and an information request expressed by means of a query we can distinguish among three subsets of documents: the set of documents which are relevant to the user's information needs, the set of documents retrieved by the query operation, also known as the answer set, and finally the intersection of these sets, namely the subset of relevant documents within the answer set.

Recall is the fraction obtained by dividing the number of relevant documents retrieved and the total number of relevant documents, and therefore is a measure of how many of the relevant documents are really retrieved.

$$R = \frac{Number\_of\_\mathrm{Re}levant\_Documents\_\mathrm{Re}trieved}{Total\_\mathrm{Re}levant\_Documents\_in\_the\_Collection} \qquad (1)$$

Precision is the fraction obtained by dividing the number of relevant documents retrieved by the total number of documents in the answer set. It can serve as a measure to determine how much of the retrieved documents do really satisfy the information needs of the user.

$$P = \frac{Number\_of\_\mathrm{Re}levant\_Documents\_\mathrm{Re}trieved}{Total\_Documents\_\mathrm{Re}trieved} \qquad (2)$$

For each query and its associated ranked list of results it is possible to construct a precision versus recall graph, which can be averaged over a series of test queries and which is then usually referred as the precision versus recall figure. This figures can be used to compare the retrieval performance over different algorithms or systems. In [Baeza 1999] Baeza-Yates and Ribeiro-Neto describe how to construct and interpolate these diagrams and they discuss some further single value summaries over precision and recall, such as the average precision at seen relevant documents, the R-precision and finally the precision histograms.

One of the main problems related to computing the recall is that it requires to have detailed knowledge of the number of relevant documents in the collection for a given query, and this is not always the possible, especially when evaluating the performance with large collections of documents.

Further measures which combine the precision and recall values are the harmonic mean proposed by Shaw et al. [Shaw 1997], computed as

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}} \qquad (3)$$

and the E evaluation measure proposed by van Rijsbergen [van Rijsbergen 1979] which in addition allows the user to tune if he is more interested in recall or in precision and which is defined as

$$E(j) = 1 - \frac{1 + b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}} \qquad (4)$$

In both cases *r(j)* is the recall for the *j*-th document in the ranking and *P(j)* is the precision for the *j*-th document in the ranking. *F(j)* and *E(j)* are respectively the harmonic mean and the E evaluation measure relative to the *j*-th document in the ranking, and in the latter case *b* is a parameter that allows the user to fine tune the measure. When *b*=1, E(j) is the complementary measure to F(j). The parameter *b* can be set to a value greater than 1 to give more relevance to precision than recall, and to a value smaller than 1 to weight more the recall than the precision.

Finally, it is interesting to note that recall and precision are objective measures that do not account for the fact that different users may have different perceptions on which are the relevant documents. To deal with this issue a new subset comprising the relevant documents known to the user is introduced to the big picture described earlier when the relevance and precision measures were presented. This allows to create new measures such as the coverage ratio, the novelty ratio, the relative recall and the recall effort [Korfhage 1997].

Reference collections

Reference collections play a fundamental role in the development and improvement of information retrieval systems. On one hand they can be used as a testbed and on the other hand, and probably more important they can be used to benchmark, as they allow that the retrieval performance of multiple systems implementing different models and architectures can be directly compared to each other.

Several standardized collections have been used throughout the years. Currently the TREC collection is considered to be the test reference collection for information retrieval. It is a result of the Text REtrieval Conference [TREC] promoted by the National Institute of Standards and Technology (NIST) with the objective of supporting research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. Within the TREC framework, a standardized large-scale collection of documents, standardized test queries and a standardized evaluation of the results generated by each of the teams is provided. This yearly retrieval conference was formerly just devoted to retrieval of text documents, but in recent years it has been extended with other tracks which include speech and video retrieval among others.

However, no specific track exists for the evaluation of music information retrieval. One of the major problems possibly relies on the fact that most of the content that would be suitable for constituting such a retrieval evaluation collection is copyright protected and the content providers are reluctant to make it publicly available. The music information retrieval research community is currently trying to build a TREC-like evaluation paradigm. Most notable, are the efforts of Downie [Downie 2003], who is for the first time building a large scale, internationally-accessible MIR testing and development infrastructure which will be hosted by the National Center for Supercomputing Applications (NCSA) at the University of Illinois. The hosted architecture will allow that a series of controlled and secured virtual research labs can be offered to the researchers and will furthermore ensure and enforce the protection of the content. Agreements are currently being reached with two content providers, HNH for over 30.000 tracks of music of their Marco Polo and Naxos record labels (approximately 3 terabytes of digital audio music information) and All Media Guide who will provide their large collection of music metadata.

2.3 Overview work and surveys on MIR

A number of overview papers, surveys and state of the art reports exist already in the area of audio and music information retrieval or on closely related topics. In 1999, Foote [Foote 1999] published the possibly first state of the art paper on audio information retrieval, covering aspects of speech recognition and identification, music and audio analysis, and finally systems and interfaces in the area of both content-based retrieval of audio and retrieval based on MIDI representations.

A further and more recent overview with a focus on the library sciences aspects is provided by Downie in [Downie 2003] where he first reviews the seven facets of music information that according to him play a relevant role in retrieval (pitch, temporal, harmonic, timbral, editorial, textual, and bibliographic), as well as the various challenges in music information retrieval (multirepresentational, multicultural, multiexperiential and multidisciplinary). Further, the "representational completeness", understood as the number of the facets listed above that are covered by a given representation, is studied on a series of systems and applications. Finally, the music information facet a brought up again and a number of systems and contributions are reviewed and implicitly classified according to the facets they use in their retrieval.

In an overview article on multimedia content analysis, Wang et. al. [Wang 2000] offer a review of some of the relevant signal processing-based fundamentals in audio content manipulation. The various frame level (short term characteristics) and clip level features are discussed, including volume, zero crossing rate, pitch and spectral features. Further some of the audio classification options are reviewed although this article puts a stronger focus on the video image content analysis.

Other relevant overview work in music information retrieval is contained within several of the PhD or Masters theses that have been presented in the field, and which should represent the current state of the art at the moment of the respective dissertations. Most relevant with regards to this particular research report are [1]:

- Downie's [Downie 1999] PhD thesis submitted in July of 1999 deals with the evaluation of a particular approach for music information retrieval in which monophonic melodies are converted to n-grams and treated as if they were words in text retrieval system. In chapter 3 of his thesis a series of relevant concepts of information retrieval systems and its evaluation are discussed, including the *Cranfield model* for retrieval evaluation and the notion of recall and precision. In addition, several previous systems are reviewed, taking into account the differentiation between analytical/production and locating musical information retrieval systems.

[1] It is important to note here that this listing of PhD thesis is not based on the actual contribution of each of the listed theses to the field of music information retrieval, but on the fact that they provide good overviews on selected aspects of MIR.

- Uitdenbogerd [Uitdenbogerd 2002] presented her PhD thesis in May of 2002 with contributions to the subject of retrieval of polyphonic performance data (MIDI). In the second chapter of her thesis, she reviews the relevant music theory elements, some fundamentals of perception such as the concepts of music memory, figure and ground, as well as the melody similarity perception taking into account its implications for music information retrieval systems. In addition, the thesis reviews computer representations of music, setting especial focus on the MIDI standard, and it goes on with an overview of different methods that have been used to compare musical pieces which each other including several similarity approaches and methods for melody extraction. Further, this chapter briefly reviews general issues of information retrieval to end with a thorough discussion of music databases and a survey on music information retrieval research.
  In addition, Uitedenbogerd's thesis contains a review on matching techniques in chapter 3. In it, she discusses the different matching algorithms or strategies as well as its implications on music information retrieval. These include exact matching, inexact matching with k errors, dynamic programming, This chapter is closed with a brief review of possible index structures.

- Blackburn [Blackburn 2000] introduces methods for content-based retrieval and navigation of music using melodic pitch contours, and within his dissertation presented in September of 2000 he reviews existing forms of representation for music content, such as the musical score, performance data (MIDI), melodic pitch contours, rhythm contours, as well as the Fourier transform, raw digital audio, and metadata related representations and standards at the time. Furthermore he reviews a number of the existing melodic retrieval systems.

- Scheirer [Scheirer 2000], submitted in April 2000 and presents methods and systems in the area of machine listening of music, and contains a good overview of on relevant research in the areas of psychoacoustics, music perception and auditory scene analysis.

- Tzanetakis' thesis [Tzanetakis 2002] submitted in 2002 contains further contributions in the field of computer audition. Each of the chapters has a section reviewing other relevant work in the field, including issues on representation, analysis, and human user interaction with audio signals and collections. Finally, and since this thesis is very near to the signal processing aspects of music information retrieval, several appendixes review relevant signal processing algorithms used within this research field, as the short time Fourier transform, the discrete wavelet transform, the mel-frequency spectral coefficients, a multiple pitch detector, a Gaussian classifier, a GMM classifier with an EM algorithm, a K-means clustering algorithm, a K nearest neighbors classifier, and the basics of principal component analysis.

- Hsu thesis [Hsu 2001] was presented in September of 2001 and deals with algorithmic and data structure aspects of content-based music information retrieval. A particular focus is set on finding repeating patterns. The first chapter contains a review of related work in music information retrieval, where Hsu concentrates on music databases and the data structures for musical object representation. A specific section is devoted to the review of contributions in the area of finding patterns for sequence data.

- Gómez [Gomez 2002] presented a research report in September 2002 which contains a clear review of the musical concepts that must be taken into account in a representation of the melody. In addition she presents a list of low lever and higher level descriptors associated to various aspects of melody representation. A further chapter includes also a review on the methods that exist for melody extraction, including algorithms for pitch estimation, note segmentation, melodic segmentation, pattern extraction in melody, and melodic transformations.

Finally, it is appropriate to note that even though pioneers have sporadically worked on it for many years, music information retrieval is a young research field as can be concluded by the fact that until the year 2000, in which the first International Symposium on Music Information Retrieval was held, no dedicated international conferences existed. The ISMIR conference [ISMIR], now in its 4th edition, has rapidly grown in contributions, participants and relevance, and is the main music information retrieval-specific research meeting.

2.4 Typical MIR applications and selected problems

From a functional point of view, the high level picture of a MIR system is clear and simple. The user posts a query which expresses certain musical information needs and the system searches among the existing documents (possibly using some indexing structures) to finally retrieve selected documents in an order of relevance to the user's information needs. Nevertheless, when it comes to the actual system design and implementation, a large number of questions arises, some of which could be:

- How to build a musically-friendly user interface for input of the query which represents the users' musical information need? Do we use text, a microphone, a CD player, …?
- In the event of using humming or whistling as query input, how to account for errors within the query, caused by a non professional technique of the user?
- Which is an appropriate intermediate representation of the musical content?
- Which of the audio properties or features should be indexed?
- By which criteria and how should the musical documents be ranked?
- What would be an appropriate visualization ("audialization") of the musical documents?
- Which are appropriate indexing structures for the musical content?
- How to build a system that can scale well with the number or size of the documents?

As we have seen earlier in this chapter, in text information retrieval generic methods and architectures have been found which work reasonably well with independence of particular characteristics or content of the text documents. In music information retrieval, however, no such universal technique has been validated yet, therefore different representations and search methods are employed depending on the application context and also especially depending on the type of content that needs to be searched. As a consequence, most of the currently existing systems in MIR have a limited scope, and they focus or even completely restrict their functionality to one content type and to a particular application area.

A first broad distinction of the content is usually made between systems that deal with polyphonic[1] music, and those which are restricted to monophonic melody sequences or monophonic sounds. In this context, we understand under polyphonic music either full songs, also polyphonic instrumental pieces as in classical music, or significant excerpts of either of them. Drum or percussion loops, although polyphonic in nature, are usually also considered separately, as different and specialized techniques are used to manipulate them.

With regards to the system types, a broad classification can be made to differentiate analytical or production systems from locating or identification systems. And further we could distinguish systems that deal with clustering or classification, identification, search (typically based on melodic or timbre similarity), visualization or structure, and with recommendation. Other distinction is frequently made based on the internal representation required by the data, symbolic or raw audio. A discussion on the most common application and its research problems follows. A complete music information retrieval system should be able to integrate all of these application scenarios into one single framework.

2.4.1 Identification applications

The purpose of an identification application is to find and locate a particular document within a data collection. Typically, the application will work with a series of features that can be extracted from a polyphonic audio signal. Given a particular music document, these features must uniquely identify that piece, although they do not necessarily have to represent any musical character of the piece. It is important that no other document should generate the same values in its features. Such applications are frequently also referred to as fingerprinting systems, as the values in the features act as a fingerprint of the musical piece [Cano 2002a] [Cano 2002b] [Batlle 2002] [Haitsma 2001] [Kalker 2001].

In addition, and in increasing order of difficulty of the problem, the features of an identification system should also be robust against variations of the signal caused by compression, distortion, noise, furthermore by variations in tempo, pitch and ultimately by slight variations of the melody or the timbre. Ideally, an identification system should be capable to detect if a given piece of music is a cover version of an already existing one. In this direction, [Yang 2002] proposes an indexing scheme and a search method which are robust against variations in the time base, under the assumption that this variation remains constant.

Typical applications of audio identification systems include the automated listening of radio broadcasts for the tracking of author's rights [Batlle 2003], or services of the telecommunications industry that allow a user to play a song over a mobile phone and

[1] polyphonic. According to Merriam-Webster; from *polyphOnos* having many tones or voices.

an automated message is returned with the name of the song and performer. In [Gomes 2003] some further applications of audio fingerprinting are discussed in the context of piracy prevention and compared to the alternative technology of watermarking.

In [Cano 2002a] the properties of an audio fingerprinting system are reviewed with regard to the requirements imposed on such a system. These include:

- **Accuracy**, as a measure of correct identifications, missed identifications and wrong identifications (also known as false positives)
- **Reliability**, will be weighted depending on the application, but false positives (as opposed to writing a title to a not identified queue) can be especially harming in copyright enforcement applications.
- **Robustness**, as the quality to accurately identify an item, independently of the amount of distortion, compression, or noise in the channel.
- **Security**, qualified as the degree of difficulty that the system can be fooled by introducing elements into the signal that would lead to an error in the identification.
- **Versatility**, in manipulating different formats, and in reusing the fingerprint database for alternative applications.
- **Scalability**, in the traditional sense, how the system scales with a growing database or with a growing number of concurrent users.
- **Complexity**, especially in the involved computations during identification time.

Further, the architecture of a fingerprinting system is described. That system relies in a method known as robust or perceptual hashing, which consists in comparing a hashed version of the incoming audio file that needs to be identified with the hashed version of the items in the collection. The hashed version is also called the signature and provides a compact representation of the audio. Musical and perceptual elements are introduced into the hashing function, so that the musical character can be preserved. In the particular system described in the paper, unsupervised training based on the Baum-Welsh algorithm is used to divide the audio into an alphabet of small units (referred to as audio description units), so that any song in the database can be described by means of a sequence of audio description units. There are 32 symbols in the used alphabet and approximately 20 per second are generated out of the audio. The identification problem is solved by applying approximate string matching to find out which of the sequences in the database is most similar to the sequence originated by the incoming audio.

2.4.2 Search by melodic similarity

According to [Selfridge-Field 1998], melody is the feature that allows a human to distinguish a work from another, and further the means that enables us to recall a tune without even knowing its text. The search of music by melody similarity is nowadays possibly one of the most active research areas within music information retrieval. It comprises problems and applications that use melody as the key for comparing documents or fragments with each other.

In musical terms, a melody can be also understood as a sequence of notes or chords. A harmonic sound produces a note which is directly related to its fundamental frequency.

As we are interested in the perception of the sound, usually the concept of pitch[1] is used instead of frequency, as besides frequency it takes into account also other factors of the sound such as sound pressure or waveform. Many psychoacoustical experiments throughout the years have shown that only a very small number of individuals have the capability to identify absolute frequencies, also called "perfect pitch", while most of the subjects demonstrate a reasonably good identification of frequency ratios between two sounds, known in music as intervals. The most important interval is the octave, corresponding to a factor of 2 in the frequency quotient. The notes in an octave interval are musically equivalent and therefore they are also given the same label. In western music, the octave is traditionally divided into 12 further intervals, called semitones, which are approximately equal in size. The actual frequency ratios for the musical intervals for the octave can vary slightly, depending on the tuning chosen. A given melody can be transposed by a number of semitones *T* up or down, meaning that each of the notes in the melody is replaced by a note *T* semitones higher or lower. Although all the underlying pitches will change, since the interval between contiguous notes in the sequence remains constant, the musical character is preserved by the transposition operation, and therefore both the original and the transposed melody can be considered equivalent.

Melody representation

Multiple methods exist for representing the melody of a musical composition. With some limitations, the score, and the Musical Instrument Digital Interface (MIDI) file format, constitute representations with a reasonable degree of accuracy. Traditionally, western composers have used the musical score to instruct musicians how to play a piece, and therefore the score includes detailed melodic information. However, some limitations can exists for representing melodies of non-western music, as musical score quantizes the melody into a set of predefined note values, corresponding to the semitones of the western musical scale. Furthermore, and in order that a melody representation can be used in a computer-based music information retrieval system it must be first converted to a digital format. Methods have been presented in the area of automatic optical character recognition (OCR) for reading in a score from a paper sheet and converting this input into MIDI, to Standard Music Description Language (SMDL), or to alternative standard file formats. [Blostein 1992] [Selfridge-Field 1994] [Bainbridge 1997] are surveys that discuss the various problems encountered in this area, also referred to as optical music recognition.

The MIDI file format is a standard interface for exchanging performance data and by design it allows for a richer representation of the melody than the score, since it overcomes the semitone quantization problem. In practice however, most of the available MIDI files are direct translations from the score and do not contain any additional melodic information with regards to the equivalent musical score. A drawback of MIDI is that some of the structural information of the music can be lost

[1] *Pitch* is defined by the American National Standards Institute (ANSI) as the "*attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from low to high. Pitch depends mainly on the frequency content of the sound stimulus, but it also depends on the sound pressure and the waveform of the stimulus.*"

[Lemström 1998]. Nevertheless, a large number of the music information retrieval systems are based on an internal MIDI representation.

Many times a computer based representation of the melody does not exist. Methods, mostly relying on the detected sequence of pitch information, have been developed for automatically generating a melody representation from the audio data. It is however a difficult problem, as cultural and perceptual factors can influence the process of automatically identifying the melodic line. Melody may have a different meaning depending on the context. It can have distinct consideration within distinct styles or types of music. Further perceptual elements that should be taken into consideration in the melody transcription process are also partially exposed in [Uitdenbogerd 2002] and [Gomez 2002]. Melody may be perceived and remembered differently by different individuals. Some audio fragments may not contain any melody, other may contain multiple overlapping melodic lines played by various voices. Furthermore, a single melodic line may be played by one voice and then switch to another (with or without overlap).

In chapter 3 of her Research Report [Gomez 2002] and in [Gomez 2003], Gómez thoroughly reviews some of the algorithms for extracting the melody from an audio signal. This task can be performed both in the temporal as well as in the frequency domains. In the time domain, the basic idea behind most methods is to determine the periodicity of the time signal. This can be easily accomplished by monitoring of the zero crossing rate or by computing the autocorrelation function, as maximal peaks will appear for a periodic signal, from which the fundamental frequency can be inferred. Further algorithms try to determine the envelope of the signal. In a periodic signal where multiple frequencies exist, the envelope will fluctuate caused by the addition of the various frequency components. The rate of the so called beatings is a function of the differences in frequency of the components; in a harmonic signal however the fundamental frequency is assumed to be dominating and thus determining the rate of the beatings. In the frequency domain, algorithms tend to be more complex, yet they appear to perform better, especially with regards to the robustness to noise, inharmonicity and peculiarities found in the spectrum. Some of these methods are further described in [Klapuri 2000] and [Maher 1993].

A popular alternative to MIDI is a note interval representation. It provides a simplification since instead of coding the actual pitch or note value, only the relative increment is annotated. This also takes advantage of the fact that musically only the interval matters, making this representation robust to transposition operations.

A further simplification of the interval-based representation consists in ignoring the relative note increment, and only accounting for the fact that the following note is equal, lower or higher in pitch. This representation is often referred to as the *melody contour*, and will reduce the alphabet to just 3 symbols. It's application can be justified based on perceptual experiments as performed by Dowling [Dowling 1978], who concludes that the contour is easier to remember by humans than the exact melody.

Comparing Melodies

The main part of the query execution phase in a system implementing search by melodic similarity consists in identifying which of the melodies in the database best match the given query. Different approaches have allowed to use different indexing schemes for efficient retrieval. Traditionally, string matching techniques on the melody representation have been used as the core technology for the comparison operation, leading to slow and non scalable systems.

Mongeau and Sankoff [Mongeau 1990] proposed a method to compute the dissimilarity of two monophonic musical sequences, which they derived from the theory of sequence comparison. It improves the basic method of *edit distance* of strings, which just accounts for insertion, deletion and replacement of a note, by including musical knowledge. In order to incorporate the notion of rhythm into the *edit distance*, it allows for replacement of several short notes by a long note and vice versa, and additionally weights the cases in which note is replaced by another of unequal length. Further, it uses the key of the melody in order to determine which note changes are towards harmonically inadequate notes and weights these higher than changes towards "harmonically compatible" notes.

In his PhD thesis, Downie [Downie 1999] introduces the concept of *musical words*, which are built by fragmenting the monophonic melody, represented as intervals, into sequences of length n, also called n-grams, and previously used in traditional information retrieval. In that work, the impact of several lengths of the n-grams and various codings (especially affecting the precision) of the interval-only representation are analyzed and evaluated. The *musical words* approach allows to apply classical information retrieval techniques on the database. The frequency of the n-grams contained in a query string can be compared to the frequencies corresponding to the n-grams of each of the indexed melodies. This statistical approach overcomes a limitation of prior systems that only indexed the musical themes or incipits.

The n-gram structure is also used for approximate matching by [Tseng 1999] and by Uitdenbogerd [Uitdenbogerd 1999], who in [Uitdenbogerd 2002] evaluated various n-gram based approaches (also ignoring the term frequency information and just counting the dissimilar n-grams) together with other techniques relying on edit distance such as local alignment or longest common subsequence to conclude that none of each methods clearly outperforms the other in a general case.

Hsu [Hsu 2001] presents approaches for coding and indexing the melody information. The rhythmic patterns in the melody are coded into *mubol strings*, and a series of operations on the mubol strings can be used to determine its rhythmic similarity distance. Further, a *music segment* is introduced as a triplet corresponding to the type (basically a shape of the melodic profile: ascending-constant-descending, descending-constant-ascending, ascending-constant-ascending and descending-constant-descending), the length of the constant segment (normalized as a fraction taking into account the preceding segment's length), and the semitone interval. A distance function for the *music segment* is also given. A suffix tree is an adequate structure for substring matching. The music segment sequence can finally be indexed with an expanded suffix tree, which was proposed by [Chen 2000] and called *augmented suffix tree*, where segment type sequence forms the structure of the suffix tree and edges are processed (and possibly split) to attach the segment duration in a first pass and the pitch information in a second pass. This allows to support efficient exact partial matching.

Finally, two very recent contributions [Shifrin 2002] and [Meek 2002] propose to use Hidden Markov Models as the basis technology for the string matching. Melodies in the database are represented in form of HMMs, where the state transition is given by the note sequence. Queries can then be considered to be observation sequences therefore the forward algorithm can be used to rank pieces according to its relevance. This method however is at a very early stage, requiring still much functional improvements especially for the case of missing notes in the query sequence, and furthermore it has only been tested on smaller collections.

Query by Humming

Many systems implementing searches by melody have been built over the last years. Among them, one particular category of application is particularly popular, as it typically implements a large chain of music information retrieval modules. In *Query by Humming* (QBH) a melody is sung, hummed or whistled by the user, preprocessed and transformed into a symbolic representation, which is then used as the query in a search by melody. An additional level of difficulty is introduced by the fact that a large number of users will introduce errors in the query due to limitations in the humming or sinning accuracy, and therefore QBH systems commonly have to incorporate methods into the query processing stage to provide an extra level of robustness. Representative QBH systems include [Ghias 1995], [Ghias 1995], or most recently [Pauws 2002].

2.4.3 Searching and browsing in timbre spaces

A further type of applications deals with the issues around classification of sounds and the browsing of these sounds in a timbre spaces. The objective in the context of content based retrieval applications is to provide new means to explore sound libraries. According to [Herrera 2003] this problem can be approached from two different angles, *perceptual description* and *taxonomic classification*. The former tries to determine the inherent properties of the sound, also called features, those which make it being perceived by humans in a particular way; the latter deals with the issues around assigning a given sound to an existing sound taxonomy. Perceptual description is therefore based on experimental results which can prove that a given feature is adequate for explaining why a sound is perceived as it is. Given a set of N features, a sound can be positioned in an n dimensional space, also referred to as the *timbre space*, and data mining techniques can be employed for finding clusters of similarly perceived sounds. In taxonomic classification, similar methods can be employed, however the objective is to determine those features that yield a maximum discrimination of the sounds.

Systems such as Soundfisher by Musclefish [Keislar 1995] or StudioOnline by IRCAM implement measures for determining the similarity of sounds. The latter implements the same timbre features that were proposed to the MPEG-7 standard and which are described in [Herrera 1999]. Given a reference sound, a list of similar sounds can be generated, based on the evaluation of a function operating on a series of selected features. The assumption is that the features and the functional computation accurately represent the perceived distance, which not always is the case. Typically, experimental systems in this application area do not scale too well. Indexing schemes would intuitively have the form of a similarity matrix, which is not practical to implement for a large number of similar sounds. Therefore, these implementations have tended to perform the similarity computation on the fly, which would not be feasible for a large number of sounds if the distance function was computationally costly.

The problem of searching elements that are not equal but close to a reference element under a given criteria, has been studied by Chávez et. al. [Chavez 2001] for the general case and under a data structures perspective. This work relies on the key underlying concept of the *metric space*, denoting a set of valid objects and a distance function that can evaluate the similarity of two elements of the set, and which has to satisfy the properties of strict positiveness, symmetry, reflexivity and most important the triangle inequality. The types of queries that are considered are range queries, nearest neighbor and k-nearest neighbor queries. A series of indexing structures and algorithms are reviewed and compared. Particularly interesting is the M-Tree [Zezula 1996], which minimizes the number of distance computations (assuming that it is a costly operation), and which also takes into account I/O performance.

2.4.4 Structure analysis, summarization and visualization

Musicologists have for a long time dealt with the structural analysis of musical pieces. An emerging application domain within MIR research deals with the automatic analysis of the structure, its summarization, and the visualization aspects of the musical information. This can apply both to music pieces, for example for highlighting with a visual metaphor encountered repeating patterns, as well as to standalone sounds, where it may be interesting to display its multiple features in an understandable way to the user. A score and the way a MIDI file is represented within a piano roll in a sequencer software are in fact two forms of visualization. Here however, we are interested in reviewing alternative transformations of audio into still images, focusing on those which may assist a user in his retrieval interaction with a MIR system.

Foote [Foote 1999b] proposes to create a self-similarity matrix of audio information, in order to display the acoustic similarity between any two instants within a recording. The matrix is constructed by windowing the audio signal in overlapping time frames and computing the first twelve mel-frequency cepstral coefficients (MFCC) plus the energy component, which appear adequate for matching similar timbres rather than exact pitches. The self-similarity value is then computed by taking a scalar product of the MFCC vectors over a time window, in order to truly capture the musical events which are of a larger duration than the initial windowing used for the MFCC computation. The self-similarity matrix is displayed as a 2 dimensional square, where both axes represent increasing time. Given a fixed time point (*i,j*) it displays the similarity between the window around instant *i* and the window around instant *j*. The coloring is selected so that the brightest pixels correspond to the highest self similarity. The matrix is symmetric, since the distance function is also symmetric, and the diagonal that constitutes the symmetry axis is a bright line since each instant should be maximally self-similar to itself. Given an audio piece with audible repeating sections or themes, these can be visualized in the matrix as bright rectangles appearing along a row (or symmetrically along a column).

In [Cooper 2002] Cooper and Foote further develop the idea of self-similarity for obtaining audio summaries. The method is capable of identifying the most representative part of a given audio from its similarity matrix. They claim that their method is independent of the media and would work equally well for video, and for time-dependent media in general.

Peeters et. al. [Peeters 2002] use an alternative approach for constructing a similarity matrix and appropriate musical summaries from a song. Their method is based on representing the signal as a sequence of successive states, obtained as a result from a classification algorithm, and which correspond to the structure of the input signal. As with humans, who would make a better summary after a second listening, this algorithm requires multiple passes in order to determine the highlights of a piece.

With a completely different purpose and approach, Rauber et.al. [Rauber 2002] use a self-organizing map as a means of hierarchically classifying and displaying music titles by its perceived sound similarity. The system architecture consists of multiple stages: preprocessing, feature extraction, analysis and visualization. In the first stage, the audio is downsampled and cut into 6 second segments, two thirds of which are discarded without further impact. The system tries to emulate a model of the human ear by extracting the loudness at different frequency bands for the duration of the segment. These features are then converted into a time invariant representation by considering first the spectrum of the amplitude modulation of the loudness, then for each of the bands 60 amplitude modulation frequency components, which are further smoothed to provide a representation of rhythmic patterns, called *modified fluctuation strength*. Titles with a strong beat in track will show a peak at a given amplitude modulation rate at all frequency bands. The modified fluctuation strength features serve as input to a growing hierarchical self-organizing map[1] (GSHSOM), a neural network algorithm that is used to map the music titles into a two-dimensional picture where similar items are grouped close to each other. In particular, the GHSOM is a hierarchy of independent SOMs where each level represents a different degree of detail.

Cano et. al. [Cano 2002c] propose yet another alternative to visually browsing by similarity in a large collection of songs. They use Fast Map, a heuristic variant of multidimensional scaling[2], to plot song objects as points on a two-dimensional graph, where the distance between songs is proportional to its distance in a feature space. The multidimensional scaling allows to reduce the original high-dimensional feature space to just 2 dimensions, maintaining however the distance metric.

2.4.6 Overview of MPEG-7

[1] Self-organizing maps, and in particular the Growing hierarchical self-organizing map, are unsupervised artificial neural networks and its explanation is out of the scope of this research work. Further information can be found in [Kohonen 1982], [Kohonen 1995]and [Dittenbach 2000].

[2] A detailed explanation of FastMap can be found in [Faloutsos 1995].

The MPEG-7 standard was released under the formal name of “Multimedia Content Description Interface” in 2002 by the Moving Picture Experts Group (MPEG), a working group of the International Organization for Standarization (ISO/IEC). MPEG acts as an international standardization body for compression, decompression, processing, and coded representation of moving pictures, audio and their combination, and has previously already developed various popular standards, including mp3 (formally MPEG-1 Layer 3).

As its name suggests, MPEG-7 provides a standard way, including the core tools and technologies, for describing multimedia content. It is particularly interesting to note that the standard only deals with content description for interchange purposes and it does not describe issues related to the coding or to the storage. In fact, other standards of the MPEG family such as MPEG-1, MPEG-2, and MPEG-4 can be used for these later purposes.

The main objective is to provide a framework to facilitate interoperability between different systems that have to work with multimedia content descriptions, including database management systems, authoring applications and query interfaces. MPEG-7 uses the concepts of Descriptors and Description Schemes to provide a series of tools to model the multimedia metadata. A Descriptor serves as an abstraction of a high or low level feature that can be annotated for a given multimedia object and provides both the syntax and the semantics of the feature representation. It can be considered analogous to the entity concept in relational data modeling. A Description Scheme is just a particular grouping of descriptors which serve as a data model for a given view or aspect of the multimedia content description.

The MPEG-7 Description Schemes cover multiple aspects of the multimedia content description and can be structured into the following categories, each of which can contain several Description Schemes:

- ***content organization*** which provides collections and models
- ***content management*** and ***content description*** including structural aspects, semantic aspects, usage, media, and creation and production Description Schemes
- ***navigation and access*** in the form of summaries, views and variations
- ***user interaction*** such as user preferences and user history
- ***basic elements***, including schema tools, basic data types, basic tools, and links and media localization elements

The standard has been defined in such a way that multiple levels of granularity can be used in the descriptions of the content. Thus, some Descriptors refer to a particular frame, others can hold the value for a whole segment, yet others can contain a summary value representing the whole content. A further property of the standard is that it allows extensibility. In the future, new Descriptors and Description Schemes may be required, therefore the standard provides a way for applications to define new such elements.

MPEG-7 is based on two W3C recommendations; the actual content descriptions are textually represented with the syntax provided by the *eXtensible Markup Language* (XML) [XML] and, although the definition of Descriptors and Description Schemes is conceptually accomplished using a *Descriptor Definition Language* (DDL) which is part of the actual MPEG-7 standard, it is in fact the tools provided XML Schema

[XMLSchema] language which are used, since XML Schema was originally developed as a means to provide definitions of the structure, the content and the semantics of XML documents.

In terms of music information retrieval, MPEG-7 does not specify how the content or even how the descriptions need to be stored. However, due to the fact that it employs XML and XML Schema, in theory, it opens the possibility to use off the shelf implementations of XML database management systems to provide a storage and retrieval framework for multimedia data, and in particular for audio and musical content.

MPEG-7 has been discussed in several publications. [Manjunath 2002] is the book written by the principal contributors to the standard, and contains detailed discussion of its main features. A further overview is provided by the various official documents of the standard, in particular by [Martinez 2002]. In [Kosch 2002] Kosch discusses the MPEG-7 standard from the perspective of multimedia database systems, to conclude that MPEG-7 has to be considered as a complement to multimedia DBMS rather than competing with. A further review of the standard is made in [van Beek 2003], however under a different perspective, as it revises which are the features of MPEG-7that allow universal metadata-driven multimedia access. In particular it reviews the elements that support personalization and usage history, as well as summarization and segmentation.

In [Lindsay 1999] and [Quackenbush 2001], the audio and music specific features of MPEG-7 are anticipated and reviewed. The former proposes various approaches for description of musical content, which can potentially serve as the basis for Descritpion Schemes in MPEG-7. The different approaches are the Annotative, the Architectural, the Medium-based, the Physical, the Perceptual and the Transcriptive. The latter contribution reviews the structure of the standard, as well as the applications which can benefit from the MPEG-7 description interface. Furthermore, [Herre 2002] presents the elements of MPEG-7 that will allow to interoperability of fingerprints generated according to the open standard specification for extraction. [Charlesworth 2000] discusses the issues that were discovered during the creation of the MPEG-7 Descriptor for describing spoken content.

Finally, a series of papers describe applications that have used the MPEG-7 standard in particular ways. Most notably in the area of MIR, are the systems developed within the CUIDADO project, some of which are described in [Wust 2004] and [Vinet 2002].

Chapter 3. Research Contributions

3.1 Introduction

This chapter presents contributions of the author to several database driven applications in the area of music information retrieval or closely related to it. A common element in all these contributions is the usage of a database to provide effective solutions to open issues in music retrieval and authoring. Most of this work has been already partially published or has been accepted for publication in conference proceedings and/or technical and scientific reports related to the CUIDADO project.

3.2 The Cuidado Project. Overview.

The acronym CUIDADO stands for Content-based Unified Interfaces and Descriptors for Audio Databases available Online. This project, partially funded by the European Commission as part of the 5th Framework Program for research on Information Society and Technology, aims at creating an infrastructure and the applications for content based manipulation and description of audio, specifically trying to support the emerging MPEG-7 standard. Detailed information on the project can be found in [Vinet 2002] or on the project's website[1].

In order to focus the research and development, as well as to have a starting point for technology validation and dissemination, two main prototype applications have driven the CUIDADO project:

- **Sound Palette**. In two versions online and the standalone offline is targeted at authoring and manipulation, and focuses on single sounds, melodies and rhythm tracks.
- **Music Browser**. Targeted at Electronic Music Distribution allows to search titles according to a series of criteria, including content-based and cultural similarities.

In addition, several low level modules related to signal processing and classification have been developed to extract relevant features for audio description. Furthermore, and described in detail in the following section, a whole database layer has been created to support the retrieval of such descriptions in MPEG-7 compatible format. Finally, it is important to note that a great effort has been devoted to designing and building appropriate interoperable architectures where all these components can be deployed minimizing interface overheads.

[1] www.cuidado.mu

3.3 Cuidado MPEG-7 database implementation

One of the activities within the CUIDADO project has consisted in analyzing the requirements and creating a corresponding database management system (DBMS) infrastructure capable of adequately addressing the needs of audio content-based retrieval applications. The starting point has been one of the industry leading database engines, Oracle, which already incorporates some basic infrastructure specifically for audio management. The InterMedia Audio component is capable of detecting common audio file formats and extracting basic audio metadata from the audio files, however in its current version it does by far not cover the high demanding requirements imposed by the MPEG-7 standard.

At this point the main issue has been to determine the best way to model the MPEG-7 standard on a database, particularly assuring that essential functionalities of the Sound Palette, a real application, are covered not only from a functional point of view, but also from a practical performance and scalability perspective. This includes providing database services that will allow to maintain the following data elements:

- editorial and authoring metadata
- structural information
- melodic and timbre information
- taxonomies and support for classification schemes

All this information has to be exchanged according to the interface provided by the MPEG-7 standard, therefore it appears reasonable to structure the data for storage as close as possible to the standard.

Technical details

A first question usually raises at the beginning of an audio content management project such as CUIDADO: whether to store the actual audio in the database or externally in files. This in fact is not too relevant from the viewpoint of content-based retrieval, as for the system to be efficient, it the audio features which have to be indexed as searches won’t occur on the audio itself but on the indexed features. Therefore there is no strict need for maintaining the audio files within database storage, and leaving the audio externally as files and storing pointers to it in the database is a perfectly valid option, which was preferred choice for the CUIDADO applications. Therefore this discussion is restricted to the storage and management of metadata.

The actual implementation is a layer on top of the Oracle 9.2. XML database management system (XMLDB), which leverages the recent advances in Oracle's XML database technology to provide MPEG-7 audio services at database management system level.

The role of the MPEG-7 layer is to complement the core XML functions provided by the standard XMLDB with the MPEG-7 specific functionalities. These functions are typically not encountered in a standard XML database implementation:

- Reference management (across documents)
- Term Reference management
- Child Node management
- Relation management
- Specific full text searches
- Classification Schemes
- Simple type extractions

This layer is conceived as an API, and it has been built as a series of pl/sql functions that encapsulate the complexities of this subsystem to the calling applications. The functions are callable directly within pl/sql, or through any of the supported proprietary or standard database driver interfaces (Native Oracle Database Drivers, JDBC, ODBC). This can be seen in Figure 1 below which displays the overall architecture. Residing inside the Oracle 9i Release 2 database, we can observe the native database components in green, the MPEG-7 (labeled MEDEE) layer on top of them, which can interact with XML-specific, but also with other object-relational elements within the database, and further up an application pl/sql layer. This pl/sql application layer is in fact an optional element that can contain application specific functionality and which can improve the performance of the system by executing all the data intensive application logic inside the database, avoiding unnecessary round-trips. Finally we can see on the top level a series of independent application modules, typically GUI applications written in C/C++ or Java, which reside externally to the database, and which can interact with the pl/SQL or with the MPEG-7 specific functions through any of the aforementioned database driver methods.

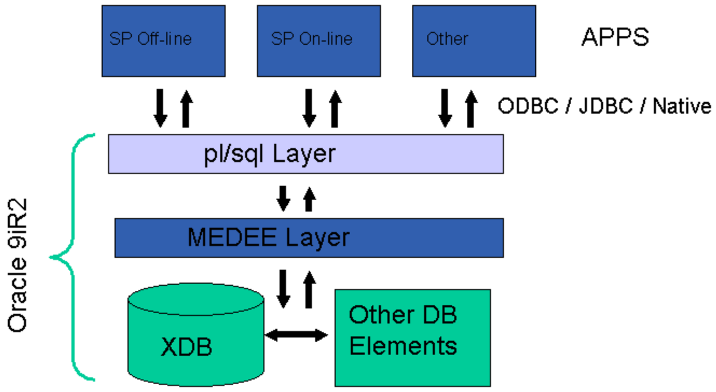


Figure 1. CUIDADO MPEG-7 database layer architecture

In order to support the MPEG-7 functionalities and especially to assure that the system scales without performance degradation the following object-relational database objects have been created:

- Tables for storing data
- Database triggers
- Stored functions and procedures
- Tables for managing function based indexes

- Indexes
- View objects

which agree to the generic schema depicted in Figure 2:

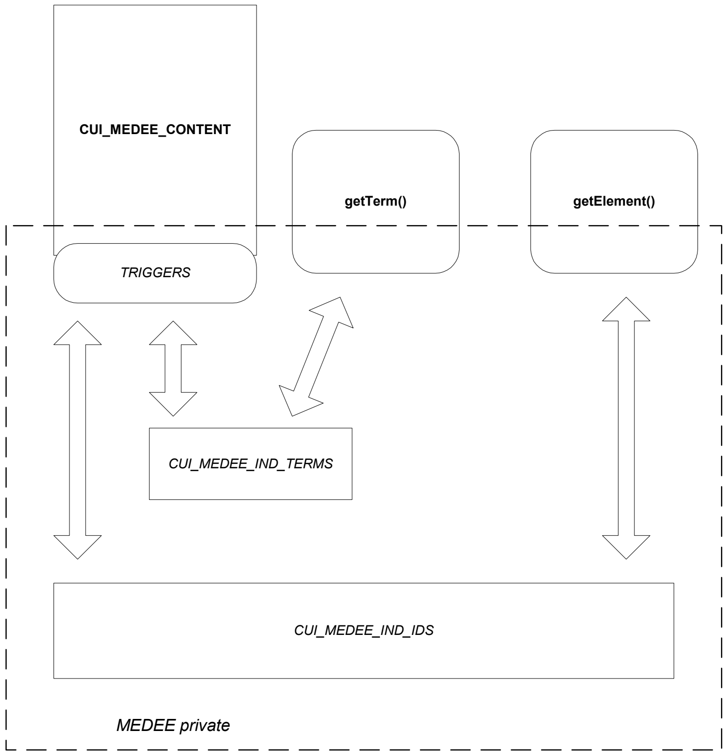


Figure 2. A logical view of the database objects that constitute the MPEG-7 layer

As can be seen in the figure, the MPEG-7 layer consists of a series of public (on top going beyond the discontinuous line) and a series of private objects. The public objects (which conform the public API) are the content table CUI_MEDEE_CONTENT and the various functions of the API (only two are shown in the diagram), which provide the MPEG-7 specific functionalities.

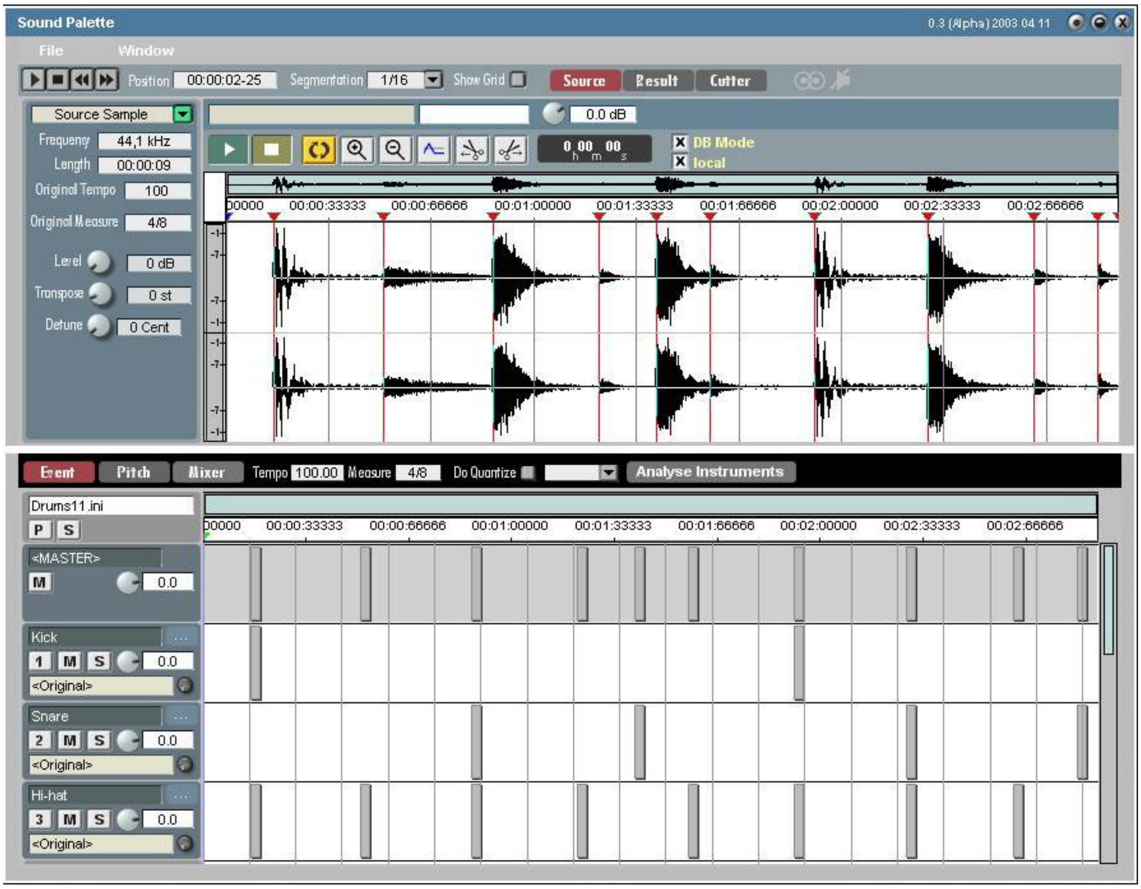
Whenever a new content is inserted, updated or deleted, a series of triggers fire to provide the desired MPEG-7 functionalities, such as MPEG-7 schema validation, consistency checking, indexing, among others. These have been coded in the CUI_MEDEE_PKG package, which makes heavy use of both XML and object relational technologies contained in the Oracle database.

This model is reusable and MPEG-7 generic, in the sense that any other new applications (besides Sound Palette online and offline) could be plugged into this architecture to obtain MPEG-7 related database services. Furthermore, since the implementation relies on the proven scalable architecture provided by the Oracle DBMS, we can expect a good scaling behavior of the MEDEE.

3.4 Cuidado Sound Palette application

The author's participation in the Sound Palette application is only indirect, by building the database infrastructures and integration modules Sound Palette relies on. However, in order to provide a better understanding on how the MPEG-7 database implementation is used by the final user a brief description of the Sound Palette application is included here. The screenshots in this section are courtesy of Creamware.

The Sound Palette represents a novel approach on audio editing and sequencing, as it offers for the first time content-based retrieval functionalities in such and environment by integrating an MPEG-7 DBMS as described in the preceding section. Furthermore, it is capable to perform content-based transformations on the loaded and imported audio files. The main graphical user interface can be seen on Figure 3 where the application displays a split screen in which a drum loop is represented. The upper window shows the stereo audio waveform, the traditional view found in most audio editors. It is interesting to notice how the loop has been automatically segmented into the individual composing sounds which is marked on the waveform by the vertical lines with a triangle on the upper end.


Sound Palette 0.3 (Alpha) 2003 04 11

Position 00:00:02-25 Segmentation 1/16 Show Grid □

Source Sample

Frequency 44,1 kHz

Length 00:00:09

Original Tempo 100

Original Measure 4/8

Level 0 dB

Transpose 0 st

Detune 0 Cent

0.0 dB

☒ DB Mode

☒ local

Tempo 100.00 Measure 4/8 Do Quantize □

Drums11.ini

<MASTER> 0.0

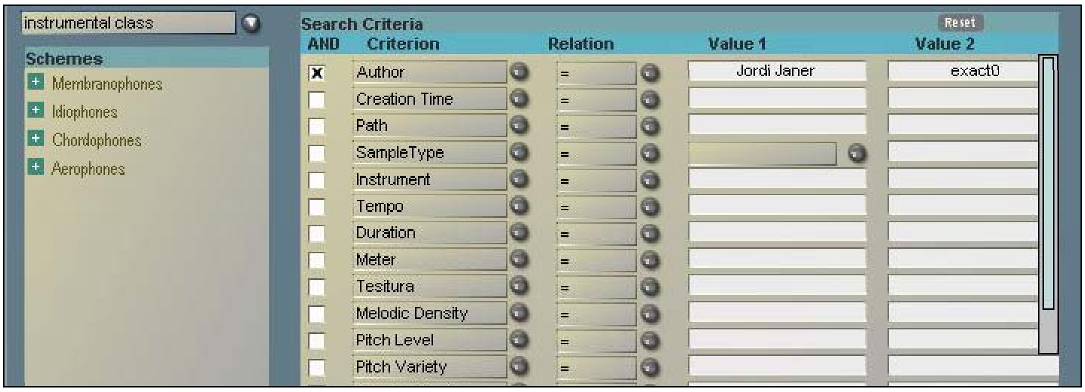
Kick 0.0 <Original>

Snare 0.0 <Original>

Hi-hat 0.0 <Original>

Figure 3. Sound Palette Off Line. Main application screen.

The lower display is showing the same drum loop, but in a format closer to how midi sequencers represent its information. In there we can see the output of the analysis, how the tool has segmented the drum loop into its composing sounds, each of which is now displayed within its own horizontal track. Each rectangle corresponds to a a triggering of the underlying sample. Most interesting from a database point of view is the fact that these samples can be substituted by other sounds from the database by using the MPEG-7 content descriptions which can be done by searching according to the available criteria or through the sound taxonomies and the Sample Browser window (Figure 4).



AND	Criterion	Relation	Value 1	Value 2
☒	Author	=	Jordi Janer	exact0
☐	Creation Time	=		
☐	Path	=		
☐	SampleType	=		
☐	Instrument	=		
☐	Tempo	=		
☐	Duration	=		
☐	Meter	=		
☐	Tesitura	=		
☐	Melodic Density	=		
☐	Pitch Level	=		
☐	Pitch Variety	=		

Figure 4. Sound Palette Offline Sample Browser

3.5 The Cuidado Music Crawler

The Music Crawler has been developed in the context of the CUIDADO project, and fits the purpose of accumulating a large number of web pages, therefore the name crawler, on which data mining techniques are applied to determine concept similarities based on the textual content found in the pages. In this particular case, the crawler is set to gather pages with musical content, the concepts are artist names, and the purpose is to determine the artist similarity from a cultural point of view. This is then used as a complementary measure in the CUIDADO Music Browser, a system that also implements timbre-based similarity of songs and browsing by musical genre.

The idea of using cultural information as a complement to signal processing measures in order to improve the results is not new, and was proposed by Pachet [Pachet 2001] and by Whitman [Whitman 2002a], [Whitman 2002b]. The former originally suggests to use co-occurrence measures and correlation analysis on radio play-list data for the purpose of classification of songs and artists into musical genres. As opposed to collaborative filtering, where subjective information is used, this method should work with objective and therefore more robust information. The latter uses data collected from sniffing the Napster-like peer to peer network OpenNap where audio files are exchanged between users and also from very specialized web pages (All Music Guide). However, it does not build a direct measure between artists, but it does indirectly through keywords (adjectives) found in relationship to the authors on the downloaded text descriptions.

The Music Crawler is based on the measures proposed in Pachet's paper and collects its data from the text contained in crawled web pages. A textual index is built to identify all the artist names found within the downloaded web pages. An artist-artist co-occurrence matrix is built, where each element *$(a_i a_{,j})$* contains the number of web pages artist a_i is found on a web page where also artist a_j appears. This matrix is symmetric and typically sparse, as ideally we should not expect web pages that would list all the artists that we are interested in considering.

In order to capture transitive similarities instead of directly using the co-occurrence as a measure for the cultural similarity, a covariance-based computation of the actual co-occurrence matrix is used. The idea is that if artists a_i and a_j have a high co-occurrence value, and if artists a_i and a_k have also a high co-occurrence value, then we could assume that a_i and a_k are to some extent culturally similar too, even in case they do not have an equally high co-occurrence value, and this behavior can be achieved by using the second order statistics.

Most of the focus in the Music Crawler design and implementation has been on obtaining a highly scalable crawler, as having a large number of web pages appears to be a prerequisite for good, objective results. In fact, a large quantity of web content can be ideally considered as an automated substitute of the human knowledge component proposed in Pachet’s paper. The initial target was set to 100 million pages, although later discarded due to the high cost in hardware and operation involved, out of the scope of the CUIDADO project. Complementary to the high scalability requirement, and due to the high cost in time and resources to crawl a large amount of web pages, the reliability of the system must be accordingly high. A data loss or inconsistencies in the data after several months of crawling cannot be acceptable. Therefore a database driven design was favored, where all the functionalities of an object relational database management system could be leveraged.

In order to support the high performance and scalability demands, a multithreaded parallel 2-tier client server architecture was designed, according to Figure 5, where we can see an array of clients which interact with the Internet and with the database server. The system supports a variable number of clients, and the database server can scale accordingly by adding more CPU power, memory and disk, and if required by adopting a cluster architecture in which multiple database server machines access a shared disk array.

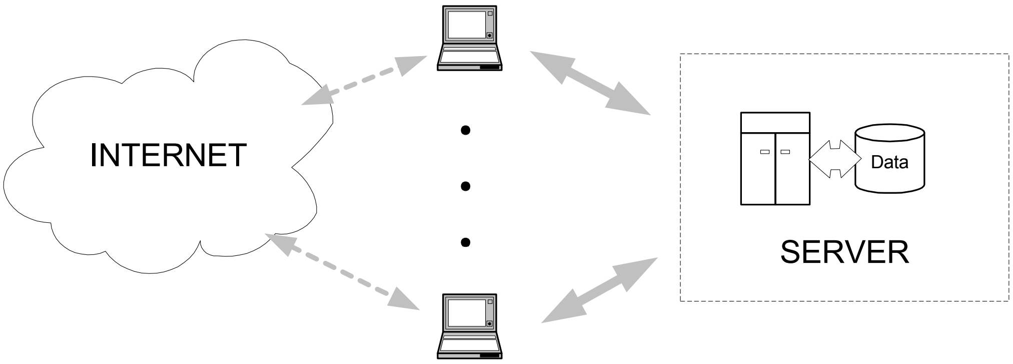


Figure 5. Architecture of the Music Crawler

Each client runs a multithreaded Java program, which executes basically 3 types of activities in independent Java threads: downloading pages, parsing downloaded pages, and URI filtering. Through a control GUI, each client can be set up to run a limited number of threads of each of the 3 types, which allows to balance the system adequately. Each thread processes one single item (usually an URI) at a time, and interacts with the database server to get its input and to deliver the output of the task. This design, in conjunction with the transactional nature of the relational DBMS, prevents any loss of data on the system, and furthermore favors a robust operation balancing all the workload among the available clients. As an example, we could imagine that one of the client PCs is switched off while parsing a given URI. That transaction will be rolled back by the database server and this same URI will be available for a different thread, possibly executing on a different client.

In support of the balancing architecture the database server implements a series of transactional work queues where the client processes can read data from and write the results to. Figure 6 shows a diagram of the data flow in the server and the various buffering queues. As we can see, each of the queues is a buffer preceding a process, some of which are executed by distributed clients in parallel, others are executed by the database server. To some extent we can consider that through the queues each of the processes is decoupled form each other. This is actually a further benefit of the architecture, as the queue architecture allows to easily move a process to a different location in the overall architecture. For instance, the URI parsing process is currently performed by a client, but it would be simple to write a new database function that does the same task, and have it read from the CUI_URI_PARSER_Q queue and write results to the CUI_URI_FILTER_Q, all this without affecting the operation of the system. Furthermore, a particular queue can be stopped, reconfigured, even reset, while the rest of the system continues its operation, which is a great advantage in terms of system maintenance.

The single queues and processes are:

- **URI Filter Process**. Performed by the Java client. Obtains a URI from the filter queue CUI_URI_FILTER_Q and decides if it is crawlable or not. Decision is based on various factors. If the URI belongs to a site that has never been crawled before, then the site's robots.txt[1] file is downloaded (if present) and the this process will determine if it has the permission to crawl the URI. In addition, this filter process will determine if this site has been banned by the Music Crawler user. This functionality allows to a priori discard sites that are considered to be not interesting. This process could execute additional filters that have currently not implemented although the Music Crawler data model would support it, such as filtering based on the depth level of the page within the site, or the number of pages already downloaded from this site, etc.

[1] Due to the nature of the http protocol, it is technically not possible to prevent a crawlers to access a given URI. However, some sites dislike being crawled and could technically ban a given IP address from the site if they suspect crawler activity. An informal protocol has been established among the crawling community and it is understood as good practice that a crawler reads a site's *robots.txt* file, which specifies which parts of that site are freely crawlable.

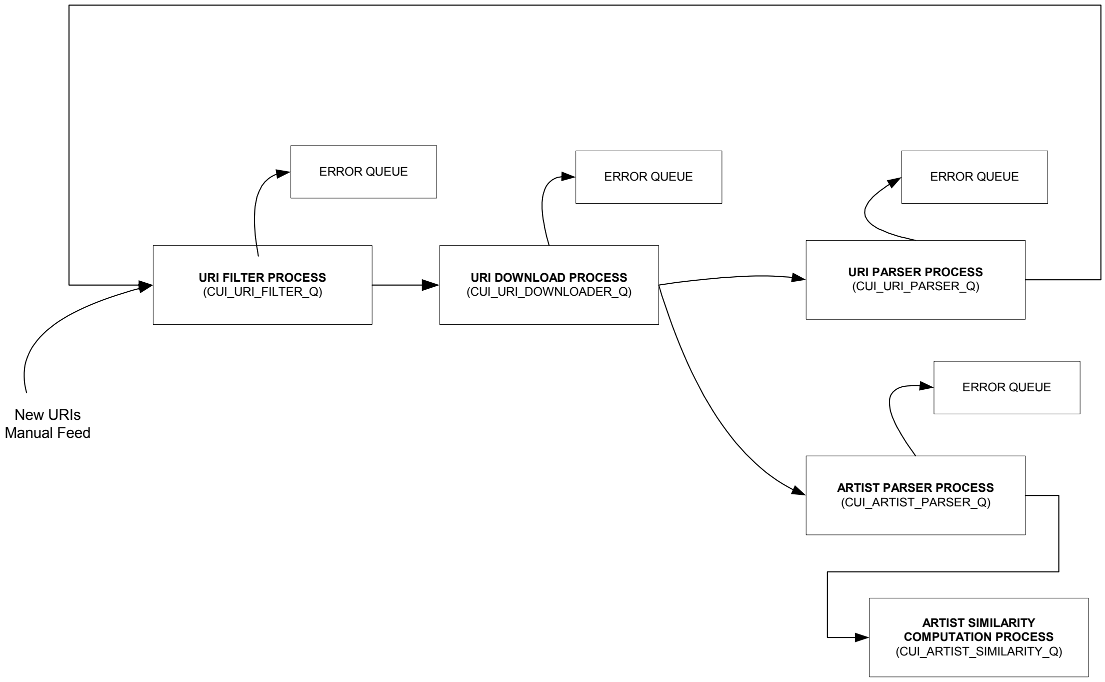


Figure 6. Logical view of database queues within the Music Crawler

- **URI Download Process**. Is heavily dependant on the available network bandwidth and also performed by a Java client. A target URI is obtained from the CUI_URI_DOWNLOADER_Q queue, a connection is opened with the destination host and URI's content is downloaded and stored in the database server. Upon completion, an URI object is written into the CUI_URI_PARSER_Q and to the CUI_ARTIST_PARSER_Q for further processing.
- **URI Parser Process**. The Java client reads from the CUI_URI_PARSER_Q queue the identifier to the URI that needs to be parsed, obtains the text from the database and does a parsing looking for further URI's within the text. At the moment only http links are followed, but recent experiments suggest that this should be extended to follow also certain JavaScript links. Any links encountered are fed back to the CUI_URI_FILTER_Q queue for further processing. The GUI offers also the possibility to manually add new URIs to the filter queue, which is necessary to start the whole crawling process. The Music Crawler tracks which of the URIs have been automatically crawled and which originate from manual input. In fact, the whole graph structure of the crawled pages is stored for allowing further future analysis and processing.
- **Artist Parser Process**. Currently this process is performed entirely in the database by means of a stored procedure. For each downloaded web page, relevant artists are identified and the URI-artist relationship is stored.
- **Artist Similarity Computation Process**. Although conceptually a queue, this is currently implemented as a standalone process which is run in the database.

All the connections from the clients to the database are established through the standard JDBC protocol. Due to the large number of clients, each of which uses a different connection for each new thread, a connection pooling mechanism has been implemented, so that connections can be reused after each transaction has been completed, and so minimizing the costly task of creating and destroying a new JDBC connection for each thread.

In the figure, additionally to each buffer queue, we can see also an error queue. These are used by each of the processes to write URIs when they have finally not been processed after repeated retries. The buffer queues are FIFO, but the "retry time" can be programmed by the administrator. As an example to illustrate this functionality, lets consider the CUI_URI_DOWNLOADER_Q queue. When an URI is read from this queue, but the downloader process times out without having read any web page (most likely because the web server is down) then the process will finish issuing a rollback statement. The URI is returned to the queue, but not dequeued again after the "retry time" has passed. This cycle can be repeated several times while the target web server is still not operational. If the operation is rolled back an Nth time (N being also a programmable variable of the queue) then instead of returning the URI to the buffer queue it will be written to the error queue.

A further element of the system that deserves particular attention is the way an artist is detected in a web page. As Pachet writes in his paper, it is not a trivial task to identify a music title or an artist from a manual annotation as there are usually many formats in which it can be written and furthermore on some names there are usually misspellings. This problem, originally formulated on the analysis of radio airplay lists written by professional DJs is heavily accentuated when we analyze web pages that are in free text format and furthermore can be written by anyone, in any language. There are even additional problems, related with the semantics of specific names. First there are artists who go by different names, as the case of Bruce Springsteen, who can be encountered as "Bruce Springsteen", "Springsteen, Bruce", "singer of the E Street band", "the Boss", … Then there are other artists which have names that are common words of the English language, such as "cake", "milk", etc. Finally there is another class of artists who have names which are really stopwords of the language, or which are reserved words as html tag names.

In order to deal with these issues the Music Crawler implements a classifier in which the artist list is expressed as a series of rules, and if any of the rules is fulfilled on a single web document, then we can say that the artist is featured in a page. So for the artist Springsteen the rules could be "(Bruce near Springsteen) OR (singer NEAR "E Street band") OR ("the Boss"). To overcome the second problem, the artist identification is performed in two steps, a first classifier tries to identify if a page is discussing musical issues or if it should be discarded by using a list of rules including words such as "music", "artist", "band", "concert", … then the actual artist filter is applied. By incorporating more sophisticated rules to the system, it can be fine tuned without further programming effort.

Although web crawling per se is not directly related to MIR, the work around the Music Crawler is clearly justified in the MIR context, and so are the earlier references by Pachet and Whitman since they are adequate means to building repositories and services that will assist content based music retrieval operations. We can easily imagine a web service or a Winamp plugin which can provide objective recommendations of similar artists or similar music titles based on web mining rather than on subjective editorial annotations. In addition, the Music Crawler presents a novel approach to crawling, by using the queue based database driven architecture exposed.

3.6 FMOL

Faust Music On-line (FMOL) is a system for real-time collaborative music composition over the web. It features a three tier database driven architecture which implements important elements for version control as well as the storage and management of the finished and in progress musical compositions. The FMOL system, originally developed by Sergi Jordà as a standalone client, has been successfully used throughout the last 5 years in multiple compositions and even life performances. FMOL allows users distributed all over the internet to work collectively on a single or on several musical pieces, sharing a common interface. It also permits new composers to modify and enlarge already existing pieces an endless number of times, while keeping at the same time the integrity of the original pieces.

Collective creation and the production of open and continuously evolving works are two of the most appealing artistic breakthroughs the Internet can offer to music composers and creators in general. The idea of musical computer networks is by no means original; earlier implementations (although on a local area scale) date back to the late 1970s with performances by the League of Automatic Music Composers (Bischoff, Gold and Horton, 1978). However, twenty years later, collective music composition or improvisation on the net, is still at a bourgeoning state and sites and projects like *Res RocketSurfer*, MIT's *Brain Opera*, William Duckworth's Internet based *Cathedral*, can probably still be counted with the fingers.

The FMOL collaborative approach is based on a vertical multitrack model (as opposed to a horizontal-exquisite corpses model, which would allow the pasting of sonic fragments one after the other). Its architecture allows each participant to start new compositions from scratch as well as overdubbing, modulating or processing any of the existing ones. FMOL has so far successfully been used as a virtual electronic music instrument for the collective composition of several scores for the Catalan theater group la Fura dels Baus, including the play F@ust 3.0 and fragments of the multimedia opera Don Quijote en Barcelona, premiered at the Gran Teatre del Liceu of Barcelona in October 2000.

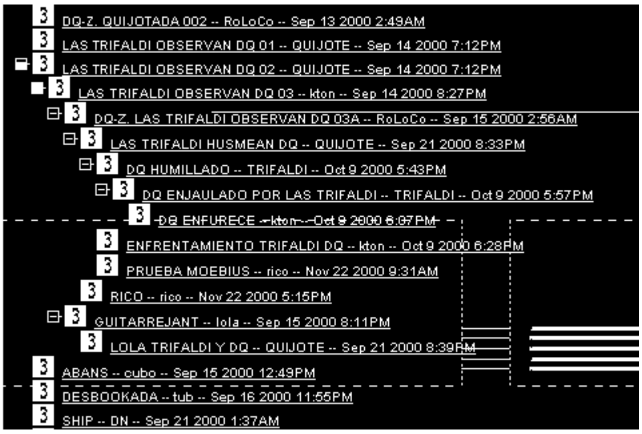
Architecture

The original system was built following a client server model. This allowed composers using the FMOL client software to log into a central server in order to download any of the pieces that were stored in a song tree structure. The composer was then able to work on some of the tracks of the piece with the standalone FMOL client, and send back the new version to the central server. Although the client server model has proven successful under specific circumstances, such as local, small sized productions, there exist disadvantages of using this architecture such as the installation process of the client, the redistribution and reinstallation of the client after a software upgrade or patch, and the inaccessibility of the database to curious Internet surfers.

The current version of FMOL has been built according to a three-tier architecture model, which has proven one of the most efficient architectures for Internet computing. The server side hosts a database server, responsible for all the storage and retrieval functions. In the middle tier an application server is responsible for executing all the application logic. The application server may be physically on the same machine as the database server or on a separate one, assuming that the network connection between both machines can support a high bandwidth and low latency.

Furthermore, such configurations will allow a high degree of scalability. If a large number of simultaneous users need to be supported, several application server machines can be set up, and connections to the system can be handled by a load balancing service, which will distribute the petitions across the application servers. Universal access to the system is guaranteed by the use of a thin client. Any Wintel personal computer equipped with a soundcard and a standard web browser will suffice for running the FMOL plug-in.

The FMOL system is based on a relational database. The main entity is the compositions table, which has a recursive relationship to itself, allowing for a tree like representation of the pieces, as shown in Figure 7. Each piece is a node storing a proprietary format score file that holds the data for eight real-time synthesized audio tracks, which can be played by the FMOL plug-in's audio engine. A user can retrieve any existing composition by clicking on the corresponding link in the tree structure, then listen to it, work on it and save a new version in the database. The new version is then be stored as a child node of the one the user picked.



- DQ-Z. QUIJOTADA 002 -- RoLoCo -- Sep 13 2000 2:49AM
- LAS TRIFALDI OBSERVAN DQ 01 -- QUIJOTE -- Sep 14 2000 7:12PM
- LAS TRIFALDI OBSERVAN DQ 02 -- QUIJOTE -- Sep 14 2000 7:12PM
  - LAS TRIFALDI OBSERVAN DQ 03 -- kton -- Sep 14 2000 8:27PM
    - DQ-Z. LAS TRIFALDI OBSERVAN DQ 03A -- RoLoCo -- Sep 15 2000 2:56AM
      - LAS TRIFALDI HUSMEAN DQ -- QUIJOTE -- Sep 21 2000 8:33PM
        - DQ HUMILLADO -- TRIFALDI -- Oct 9 2000 5:43PM
          - DQ ENJAULADO POR LAS TRIFALDI -- TRIFALDI -- Oct 9 2000 5:57PM
            - ~~DQ ENFURECE -- kton -- Oct 9 2000 6:07PM~~
          - ENFRENTAMIENTO TRIFALDI DQ -- kton -- Oct 9 2000 6:28PM
          - PRUEBA MOEBIUS -- rico -- Nov 22 2000 9:31AM
        - RICO -- rico -- Nov 22 2000 5:15PM
    - GUITARREJANT -- lola -- Sep 15 2000 8:11PM
      - LOLA TRIFALDI Y DQ -- QUIJOTE -- Sep 21 2000 8:39PM
- ABANS -- cubo -- Sep 15 2000 12:49PM
- DESBOOKADA -- tub -- Sep 16 2000 11:55PM
- SHIP -- DN -- Sep 21 2000 1:37AM

Figure 7. Faust Music Online compositions tree

A common problem in collaborative composition systems is intellectual property rights tracking. In this case, one of the design purposes has been to allow global access to musical collaboration. As a result many participating composers are casual Internet surfers, which makes this control even more difficult. The FMOL system implements a rights tracking option, which requires that a user is registered before allowing changes to the songs database. This control was used on the system first implementation, in 1998, after an agreement with the Spanish authors' association, SGAE, who sponsored the project and facilitated all the registration proceedings even for non associated authors. This feature has not been used, however, in the latest implementation.

Figure7 shows a fragment of the compositions tree. Each line represents a node and displays the title of the piece, the author's alias and creation date. The amount of indentation reflects the depth or number of generations of the piece. In this case, several users have interacted to create up to 7 layers of collaborative work, and some of the layers (i.e. 3, 4 and 5) have different siblings. Users are also allowed to vote on the quality of any composition. This information can help in the final selection process, and can be used to build a profile of the users of the system. The system allows also to retrieve several top 10 rankings, as th last 10 accessed, a ranking based on collaborative filtering of the voting, as well as a search by title or author.

The middle tier hosts the application server and the web server. These software components are responsible for running most of the program logic of the FMOL system as well as serving the presentation layer to the web browser. This includes the dynamic generation of all the web pages for user registration, profiling, voting, and most important, displaying the composition trees and managing the upload and download of compositions.

The client tier is said to be thin because it only consists of a browser running a plug-in. The application logic is hosted mostly in the middle tier leaving the client layer only for the synthesis engine, the graphical interface and the presentation logic. Despite the design objective of keeping the software running on the client to a minimum, there were both important esthetical and social reasons for including a specific proprietary synthesis engine, as one of the main objectives of the project was to approach experimental electronic music creation to newcomers and hobbyist musicians. In that sense, the FMOL composition and synthesis plug-in grants that everybody has access to compose, even surfers without any other audio software and no more hardware than a multimedia soundcard. This enforces an equal opportunity environment, while forcing at the same time, real-time composition and sound manipulation by means of innovative and intuitive graphical interfaces.

Although this three-tier architecture allows for different approaches which may be applied in the future without loosing any generality (as for instance the use of standard MIDI files or any other standard format which could be generated with currently available and generic software and without the need for a specific synthesis plug-in), the current synthesizer engine architecture and its graphical interfaces were in fact specially designed with this collaborative approach in mind.

The engine, written in C++ for the wintel platform, was meant to be a complete sound generation kernel flexible enough for real time synthesis and processing on a low-end machine (e.g. Pentium 200), that could be appealing and enriching for users with different skills and electronic music knowledge. The current version supports eight stereo real-time synthesized audio channels or tracks, each consisting of a generator (sine, square, Karplus-Strong, sample player, etc.) and three serial processors (filters, reverbs, resonators, ring-modulators, etc.) to be chosen by each composer between more than a hundred different synthesis methods or algorithms.

Most important, this architecture allows any composer not only to add new sound layers to previous compositions, but also to apply further processing to any of the composition’s existing tracks, modulating or distorting what other composers did, in unpredictable manners. That way, a musical idea brought by one composer can grow and evolve in many different directions unexpected by its original creator.

Chapter 4. Conclusions and future research interests

This work has reviewed some of the relevant issues in the field of information retrieval, specifically in the area of content based retrieval of music. Furthermore it has illustrated some of the typical applications of music information retrieval at a scientific and at a technical level. A large number of contributions have been referenced, and possibly the number of relevant contributions to specific aspects in this field that have not been referenced in this work might be even larger.

Despite all these successful research efforts in the field of music information retrieval, a large gap still appears to exist between what can nowadays be offered at scientific or algorithmic level and what has been made available to the general public in the form of applications and usable functionalities. And the reason does not seem to be a lack of commercial interest, but rather the fact that many questions remain open and require further technical and scientific clarification.

Even though we can extract many low level features from an audio signal, how precisely do they map to what is perceived by a user? How must that mapping be modified so that it suits another user who may be less trained in music listening? How are these mapping functions implemented in a search engine, and which data structures will best suite these functions for efficient retrieval? Which scalable algorithms should be employed?… This list could go on and on.

There are other questions that go beyond a strictly technical level and which also seem appealing for scientific study. In the current scenario we have, for the first time, a set of low level tools to perform acceptable content based manipulation and retrieval tasks. The amount of stored multimedia information seems to be increasing at rates even higher than Moore's law. Can we expect the death of the current file system in favor of database oriented content management solutions? And questioning even further we could inquire about socio-economical an behavioral aspects. How will the music information retrieval technology (in conjunction with other collateral advances as universal and ubiquitous internet access, portable devices, peer to peer networks and grid computing, …) impact the way music is listened to, distributed and commercialized? Given that framework, which novel services and applications could be provided and which technologies will be needed to make them effective?

By taking a look at the titles of the accepted papers and posters for the forthcoming ISMIR 2003 conference, it is possible to deduce that most of the topics are still hot, with the possible exception of song identification, a problem that appears to be solved. Furthermore, it is interesting to note the growth of contributions in the area of music information retrieval evaluation.

To conclude this work a brief list of topics follows which summarize several interests of the author in further work on the area of database-driven systems and applications for music information retrieval and retrieval:

- **MPEG-7 databases:** Database implementation of distance functions for similarity computation. Indexing structures and scalable algorithms.

- **Cultural similarity:** Accumulation of "cultural" data on music and mining techniques to deduce similarities measures that can be used as complement of a signal-based similarity. Mining of the web and peer to peer networks.
- **New applications and new visual interfaces:** Analysis of new interfaces as result of database queries. Relevance feedback in music information retrieval.
- **Socio-economic impacts:** Analysis of the socio-economic impact, transformation and consequences of content based manipulation and retrieval of music.

References

[Baeza 1996] Ricardo Baeza-Yates, Gonzalo Navarro. Integrating contents and structure in text retrieval. ACM SIGMOD Record, Volume 25 Issue 1. March 1996.

[Baeza 1999] Ricardo Baeza-Yates, Berthier Ribeiro-Neto. Modern Information Retrieval. ACM Press Addison Wesley, 1999.

[Bainbridge 1997] Bainbridge, D. and N. Carter, Automatic Recognition of Music Notation, Handbook of Optical Character Recognition and Document Image Analysis, H. Bunke and P. Wang (eds), World Scientific, Singapore, 1997, pp. 557-603.

[Batlle 2002] Batlle, E. Masip, J. Guaus, E. Automatic Song Identification in Noisy Broadcast Audio. Proceedings of IASTED International Conference on Signal and Image Processing 2002. Kauai, Hawaii, USA. 2002.

[Batlle 2003] Batlle, E. Masip, J. Cano, P. System analysis and performance tuning for broadcast audio fingerprinting. Proceedings of 6th International Conference on Digital Audio Effects. London, UK. 2003.

[Blackburn 2000] Steven George Blackburn. Content Based Retrieval and Navigation of Music Using Melodic Pitch Contours. PhD Thesis. University of Southampton. September 2000.

[Blostein 1992] Blostein, D. and Baird, H. S. A Critical Survey of Music Image Analysis. Structured Document Image Analysis, pp. 405-434. Springer-Verlag, Berlin. 1992.

[Burkowski 1992] Forbes J. Burkowski. Retrieval activities in a database consisting of heterogeneous collections of structured text. Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. June 1992.

[Cano 2002a] Cano, P. Gómez, E. Batlle, E. Gomes, L. Bonnet, M. Audio Fingerprinting: Concepts and Applications. Proceedings of 2002 International Conference on Fuzzy Systems Knowledge Discovery. Singapore. 2002.

[Cano 2002b] Cano, P. Batlle, E. Mayer, H. Neuschmied, H. Robust Sound Modeling for Song Detection in Broadcast Audio. Proceedings of 112th AES Convention, 2002. Munich, Germany. 2002.

[Cano 2002c] Cano, P. Kaltenbrunner, M. Gouyon, F. Batlle, E. On the use of Fastmap for audio information retrieval and browsing. Proceedings of ISMIR 2002 - 3rd International Conference on Music Information Retrieval. Paris, France. October 2002.

[Charlesworth 2000] J. P. A. Charlesworth, P. N. Garner. Spoken content metadata and MPEG-7. ACM Multimedia Workshops 2000: 81-84

[Chavez 2001] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates and José Luis Marroquín. Proximity Searching in Metric Spaces. ACM Computing Surveys 33(3):273--321, 2001.

[Chen 2000] Arbee L. P. Chen, Maggie Chang, Jesse Chen, Jia-Lien Hsu, Chih-How Hsu, Spot Y. S. Hua. Query by Music Segments: An Efficient Approach for Song Retrieval. IEEE International Conference on Multimedia and Expo 2000. pp 873-876. 2000.

[Cooper 2002] Matthew Cooper and Jonathan Foote. Automatic Music Summarization via Similarity Analysis. Proc. Third International Symposium on Musical Information Retrieval (ISMIR), Paris, October 2002.

[Crestani 1998] Fabio Crestani, Mounia Lalmas, Cornelis J. Van Rijsbergen, Iain Campbell. Is this document relevant?…probably: A survey of probabilistic models in information retrieval. ACM Computing Surveys (CSUR), Volume 30 Issue 4. December 1998.

[Date 1994] C. Date. An Introduction to Database Systems. Addison-Wesley, 6th edition, 1994.

[Dittenbach 2000] M. Dittenbach, D. Merkl, and A. Rauber. The Growing Hierarchical Self-Organizing Map. In Proceedings of the International Joint Conference on Neural Networks 2000 (IJCNN'2000), 24. - 27. 7. 2000, Como, Italy, 2000.

[Downie 1999] Downie, J. Stephen. Evaluating a Simple Approach to Music Information Retrieval: Conceiving Melodic N-grams as text. PhD Thesis. The University of Western Ontario.London, Ontario. July, 1999.

[Downie 2003] Downie, J. Stephen. Music information retrieval (Chapter 7). In Annual Review of Information Science and Technology 37, ed. Blaise Cronin, 295-340. Medford, NJ: Information Today, 2003.

[Downie 2003] J. Stephen Downie. The TREC-Like Evaluation of Music IR Systems. To appear in Proceedings of SIGIR2003.

[Downling 1978] W.J. Dowling. Scale and contour. Two components of a theory of memory for melodies. Psychological Review 85(4); 341-354, 1978.

[Faloutsos 1995] C. Faloutsos and K. Lin. FastMap: A Fast Algorithm for Indexing, DataMining and Visualization of Traditional and Multimedia Datasets. In Proceedings of the 1995 ACM SIGMOD, 163-174,1995.

[Foote 1999] J. Foote, An Overview of Audio Information Retrieval, Multimedia Systems, Vol. 7, No. 1, ACM Press/Springer-Verlag, January 1999.

[Foote 1999b] J. Foote. Visualizing Music and Audio using Self-Similarity. In Proceedings of ACM Multimedia '99, (Orlando, FL) ACM Press, pp. 77-80, 1999

[Foote 2002] Jonathan Foote, Matthew Cooper, and Unjung Nam. Audio Retrieval by Rhythmic Similarity. Proc. Third International Symposium on Musical Information Retrieval (ISMIR), Paris, October 2002.

[Frakes 1992] W.B. Frakes and R. Baeza-Yates. Information Retrieval: Data Structures & Algorithms. Prentice Hall, Englewood Cliffs, NJ, USA, 1992.

[Furnas 1988] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, K. E. Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval. May 1988.

[Ghias 1995] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. Query by humming: musical information retrieval in an audio database. Proceedings of ACM Multimedia Conference 1995, San Francisco, California, November 1995.

[Gomes 2003] Gomes, L. Cano, P. Gómez, E. Bonnet, M. Batlle, E. Audio Watermarking and Fingerprinting: For Which Applications?. Journal of New Music Research Vol.32 .1. 2003

[Gomez 2002] Emilia Gómez. Melodic Description of Audio Signals for Music Content Processing. Doctoral Pre-Thesis Work. UPF. Barcelona, 2002.

[Gomez 2003] Gómez, E. Klapuri, A. Meudic, B. Melody Description and Extraction in the Context of Music Content Processing. Journal of New Music Research Vol.32 .1. 2003.

[Haitsma 2001] J. Haitsma, T. Kalker, and J. Oostveen. Robust Audio Hashing for Content Identification. Proceedings of the International Workshop on Content-Based Multimedia Indexing, Brescia, Italy, Sept. 2001.

[Herre 2002] J. Herre, O. Hellmuth, M. Cremer. Scalable Robust Audio Fingerprinting Using MPEG-7 Content Description. IEEE MMSP Workshop on Multimedia Signal Processing, St. Thomas, USA 2002.

[Herrera 1999] Herrera, P. Serra, X. Peeters, G. Audio Descriptors and Descriptor Schemes in the Context of MPEG-7. Proceedings of International Computer Music Conference 1999. Beijing, China.

[Herrera 2003] Herrera, P. Peeters, G. Dubnov, S. Automatic Classification of Musical Instrument Sounds. Journal of New Music Research Vol.32 .1. 2003.

[Hollaar 1976] L. A. Hollaar, B. J. Hurley, D. J. Kuck, D. H. Lawrie, J. W.S. Liu, J. M. Milner, J. K. Morgan, J. R. Rinewalt, W. H. Stellhorn. The design of system architectures for information retrieval. Proceedings of the annual conference. October 1976.

[Hsu 2001] Jia-Lien Hsu. Content-based Music Information Retrieval and Analysis. PhD Thesis. National Tsing Hua University. September 2001.

[ISMIR] www.ismir.net

[Kalker 2001] T. Kalker. Applications and Challenges for Audio Fingerprinting. Presentation at the 111th AES Convention, New York, 2001.

[Keislar 1995] Keislar, D., Blum, T., Wheaton, J., & Wold, E. Audio analysis for content-based retrieval. Proceedings of the 1995 International Computer Music Conference, (pp. 199-202). San Francisco, CA: International Computer Music Association. 1995

[Klapuri 2000] A. Klapuri. Qualitative and quantitative aspects in the design of periodicity estimation algorithms. In European Signal Processing Conference, 2000.

[Kohonen 1982] T. Kohonen. Self-organized formation of topologically correct feature maps. Biological Cybernetics. 43:59-69, 1982.

[Kohonen 1995] T. Kohonen. Self-organizing maps. Springer Verlag, Berlin, 1995.

[Kosch 2002] H. Kosch. MPEG-7 and Multimedia Database Systems. ACM SIGMOD Record, Vol 31, Issue 2, June 2002.

[Lemstrom 1998] Lemström, K., & Laine, P. Musical information retrieval using musical parameters. In Proceedings of the International Computer Music Conference. 1998.

[Lesk 1997] Michael Lesk. Practical Digital Libraries; Books, Bytes, and Bucks. Morgan Kaufmann, 1997.

[Lindsay 1999] Adam Lindsay and Werner Kriechbaum. There's more than one way to hear it: Multiple representations of music in MPEG-7. Journal of New Music Research, 28 (1999), No. 4, pp. 364-372. 1999.

[Maher 1993] R. C. Maher and J. W. Beauchamp. Fundamental frequency estimation of musical signals using a two-way mismatch procedure. Journal of the Acoustic Societyof America, 95:2254-2263, 1993.

[Manjunath 2002] B. S. Manjunath, P. Salembier, T. Sikora. Introduction to MPEG-7: Multimedia Content Description Interface. Wiley, April 2002.

[Maron and Kuhns 1960] Maron, M. E. AND Kuhns, J. L. On relevance, probabilistic indexing and retrieval. J. ACM 7, 216-244.1960.

[Martinez 2002] MPEG-7 Overview (version 8). ISO/IEC JTC1/SC29/WG11 N4980, Klangenfurt. July 2002.

[McNab 1996] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham. Towards the digital music library: Tune retrieval from acoustic input. in Digital Libraries Conference, 1996.

[Meek 2002] Meek, C. and W.P. Birmingham. Johnny Can't Sing: A Comprehensive Error Model for Sung Music Queries. in ISMIR 2002. Paris, France. 2002.

[Meghini 2001] Carlo Meghini, Fabrizio Sebastiani, Umberto Straccia. A model of multimedia information retrieval. Journal of the ACM (JACM), Volume 48 Issue 5. September 2001.

[Mongeau 1990] M. Mongeau, D. Sankoff. Comparison of Musical Sequences. Computers and the Humanities 24, 1990, 161-175. 1990.

[MPEG-7] http://www.mpeg-industry.com

[Ogawa 1991] Yasushi Ogawa, Tetsuya Morita and Kiyohiko Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and a learning method. Fuzzy Sets and Systems. Volume 39, Issue 2.January 1991.

[Ozkarahan 1999] Esen Ozkarahan, Fazli Can. Multi-media document representation and retrieval. Proceedings of the 19th annual conference on Computer Science. April 1999.

[Pachet 2001] Pachet, F., Westerman, G. and Laigre, D. Musical Data Mining for Electronic Music Distribution. Proceedings of the 1st WedelMusic Conference, 2001.

[Pauws 2002] Steffen Pauws. CubyHum: A Fully Operational Query by Humming System. Proceedings of ISMIR 2002, Paris, October 2002

[Peeters 2002] Geoffroy Peeters, Amaury La Burthe, Xavier Rodet. Toward Automatic Music Audio Summary Generation from Signal Analysis. Proc. Third International Symposium on Musical Information Retrieval (ISMIR), Paris, October 2002.

[Quackenbush 2001] Schuyler Quackenbush and Adam Lindsay. Overview of MPEG-7 Audio. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, June 2001.

[Rauber 2002] A. Rauber, E. Pampalk, and D. Merkl. Using Psycho-Acoustic Models and Self-Organizing Maps to Create a Hierarchical Structuring of Music by Sound Similarities. Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02), pp 71-80, Paris, France, October 13-17, 2002.

[Ribeiro-Neto 1996] Berthier A. N. Ribeiro, Richard Muntz. A belief network model for IR. Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. August 1996.

[Robertson and Sparck Jones 1976] Robertson, S. E. and Spark Jones, K. Relevance weighting of search terms. J. Am. Soc. Inf. Sci. 27, 129-146. 1976.

[Salton 1982] G. Salton, C. Buckley, C. T. Yu. An evaluation of term dependence models in information retrieval. Proceedings of the 5th annual ACM conference on Research and development in information retrieval. May 1982.

[Salton 1983] G. Salton and M. J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill Book Co., New York, 1983.

[Salton 1983b] Gerard Salton, Edward A. Fox, Harry Wu. Extended Boolean information retrieval. Communications of the ACM, Volume 26 Issue 11. November 1983.

[Scheirer 2000] Eric D. Scheirer. Music-Listening Systems. PhD Thesis. Massachusetts Institute of Technology. June, 2000.

[Selfridge-Field 1994] Selfridge-Field, E. Optical recognition of music notation: A survey of current work. Computing in Musicology: An International Directory of Applications, Hewlett, W. B. and E. Selfridge-Field (eds), Thomas-Shore, Michigan, 1994, vol. 9, pp. 109-145.

[Selfridge-Field 1998] Selfridge-Field, E. Conceptual and representational issues in melodic comparison. In Melodic Similarity - Concepts, Procedures, and Applications, Hewlett,W.B. & Selfridge-Field, E. editors, MIT Press, Cambridge, Massachusetts. 1998.

[Shaw 1997] William M. Shaw, Jr., Robert Burgin, and Patrick Howell. Performance standards and evaluations in IR test collections: Vector-space and other retrieval models. Information Processing and Management, 33(1):15-36, 1997.

[Shifrin 2002] Shifrin, J., B. Pardo, and W. Birmingham. HMM-Based Musical Query Retrieval. in Joint Conference on Digital Libraries. 2002. Portland, Oregon. 2002.

[Subrahmanian 1999] V.S. Subrahmanian. Principles of Multimedia Database Systems. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.

[TREC] http://trec.nist.gov

[Tseng 1999] Tseng, Y. H. Content-based retrieval for music collections. In Proceedings of ACM SIGIR. 1999.

[Turtle 1989] H. Turtle, W. B. Croft. Inference networks for document retrieval. Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval. December 1989.

[Turtle 1991] Howard Turtle, W. Bruce Croft. Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems (TOIS), Volume 9 Issue 3. July 1991.

[Tzanetakis 2002] George Tzanetakis. Manipulation, Analysis and Retrieval Systems fro Audio Signals. PhD Thesis. Princeton University. June 2002.

[Uitdenbogerd 1999] A.L. Uitdenbogerd, and J. Zobel. Matching Techniques for Large Music Databases. Proceedings of the ACM Multimedia Conference, Orlando, Florida, 57-66, 1999.

[Uitdenbogerd 2002] A.L. Uitdenbogerd, and J. Zobel, Music ranking techniques evaluated. Proc. Australasian Computer Science Conference, Melbourne, Australia, 275-283, 2002.

[Uitdenbogerd 2002] Alexandra L. Uitdenbogerd. Music Information Retrieval Technology. PhD Thesis. RMIT University. Melbourne. May 2002.

[Ullman 1982] J. D. Ullman. Principles of Database Systems, 2nd Edition. Computer Science Press, 1982.

[van Beek 2003] Peter Van Beek, John R. Smith, Touradj Ebrahimi, Teruhiko Suzuki and Joel Askelof. Metadata Driven Multimedia Access, IEEE Signal Processing Magazine, Special Issue on Universal Multimedia Access, vol. 20(2):40-52, March 2003.

[van Rijsbergen 1979] C. J. van Rijsbergen. Information Retrieval. Butterworths, 1979.

[Vinet 2002] H. Vinet, P. Herrera, F. Pachet, "The CUIDADO Project", Proceedings of the 3rd International Conference on Music Information Retrieval, Paris, October 2002.

[Wang 2000] Yao Wang, Zhu Liu, and Jin-Cheng Huang . Multimedia Content Analysis (Using Both Audio and Visual Clues). IEEE Signal Processing Magazine, November 2000.

[Weber 1999] R. Weber, J. Bollinger, T. Gross, H.J. Schek. Architecture of a networked image search and retrieval system. Proceedings of the eighth international conference on Information and knowledge management. November 1999.

[Whitman 2002a] Whitman, Brian and Steve Lawrence. Inferring Descriptions and Similarity for Music from Community Metadata. Proceedings of the 2002 International Computer Music Conference. pp 591-598. 16-21 September 2002, Göteborg, Sweden.

[Whitman 2002b] Whitman, Brian and Paris Smaragdis. Combining Musical and Cultural Features for Intelligent Style Detection. Proceedings of the 3rd International Conference on Music Information Retrieval. Paris, France. October 2002.

[Witten 1994] I.H. Witten, A. Moffat, and T.C. Bell. Managing Gigabytes: Compressing and Indexing Documents and Images. Van Nostrand Reinhold, New York, 1994.

[Wong 1985] S. K. M. Wong, Wojciech Ziarko, Patrick C. N. Wong. Generalized vector spaces model in information retrieval. Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval. June 1985.

[Wust 2004] Otto Wüst. An MPEG-7 Database System and Application for Content-Based Management and Retrieval of Music. To appear in IEEE Proc. of the 10th International Multimedia Modelling Conference. 2004.

[XML] http://www.w3.org/XML/

[XMLSchema] http://www.w3.org/XML/Schema

[Yang 2002] Cheng Yang. Efficient Acoustic Index for Music Retrieval with Various Degrees of Similarity. In Proc. ACM Multimedia, 2002.

[Zezula 1996] P. Zezula, P. Ciaccia, and F. Rabitti. M-tree: A dynamic index for similarity queries in multimedia databases. Technical Report 7, HERMES ESPRIT LTR Projects, 1996.

Annex

This annex section contains the following papers, written or co-written by the author:

- Otto Wüst. An MPEG-7 Database System and Application for Content-Based Management and Retrieval of Music. To appear in IEEE Proc. of the 10th International Multimedia Modelling Conference. 2004.

- Sergi Jordà, Otto Wüst. A System for Collaborative Music Composition over the Web. IEEE Proc. 12th International Conference on Database and Expert System Applications. Munich, Germany. September 2001.

An MPEG-7 Database System and Application for Content-Based Management and Retrieval of Music

Otto Wüst
Music Technology Group, Audiovisual Institute, Pompeu Fabra University
Passeig de la Circumval·lació, 8, 08003 Barcelona, Spain
email: otto.wust@iua.upf.es

Abstract

Computer users are gaining access to and are starting to accumulate moderately large collections of multimedia files, in particular of audio content, and therefore demand new applications and systems capable of effectively retrieving and manipulating these multimedia objects in novel ways.

Content-based retrieval of multimedia files is typically based on searching within a feature space, defined as a collection of parameters that have been extracted from the content and that describe it in a relevant way for that particular retrieval application. The MPEG-7 standard offers the tools to model these meta data in an interoperable and extensible way, and can therefore be considered as a new framework for building content-based audio retrieval systems.

This paper explains how an MPEG-7 compatible content-based music retrieval database management system has been designed and built. The author puts a special focus on both the relevant data model considerations for music information retrieval as well as on the particular database architecture for supporting the MPEG-7, XML-Schema based standard. Finally the author briefly describes an audio application that makes use of this content-based retrieval system, which serves as a proof that novel content-based music applications can indeed be successfully architected with the database technologies currently available.

1. Introduction

Large amounts of digital audio are nowadays widely available. This can range from musical pieces or songs for download over the internet by a casual computer user, to highly specialized sound libraries delivered on DVD-ROM targeted for the professional musician.

As the number of available media increases, so does the necessity for a user to locate the desired audio files in an efficient way. The field of information retrieval deals with the modeling, the indexing and access of information mainly within digital libraries. It has been widely studied for many years [1], mostly focusing on retrieving textual information using text-based methods. In recent times, the growth of the Internet and the general availability of multimedia content, have lead to new specialized research areas of information retrieval, that either focus on the new Internet medium [2], the inherent properties of the multimedia elements to be retrieved as is the case of audio and video information retrieval [3][4], or on particular data manipulation techniques, such as data mining in order to extract or derive information otherwise hidden in the data [5][6].

Database management systems (DBMS) have been widely used to implement efficient text based information retrieval systems, solving many of the problems encountered in that field. However, in the area of multimedia, and in particular in audio and music information retrieval, there is still a lot of ongoing research and there are open questions concerning architectural aspects such as scalability of the systems, as well as functional issues related to the usability.

This paper addresses the possibility of using available DBMS as the key building block for music retrieval applications that can satisfy the needs of the modern user. In the next section the author elaborates a list of aspects related to the modeling of musical data, which should be taken into consideration when designing a database for content-based manipulation of musical information. Then, the MPEG-7 standard [7] is reviewed and aspects that may have impact using it for developing a music retrieval system are highlighted. The description of a real MPEG-7 database implementation for content-based processing of MPEG-7 descriptions follows. In addition, an example novel content-based audio editor, manipulation, and authoring application, which makes exhaustive use of the database features, is presented and conclusions are drawn.

2. Relevant considerations on modeling a database system for retrieval of musical content

The scope of the term musical content is not precisely defined. It could range from text documents that contain music-related information (any text format including hyperlinked web pages), to symbolic representations of music (like sheet music and MIDI files) and multimedia files (audio and video). In the context of multimedia modeling and content-based multimedia information retrieval, we can restrict the problem of querying for musical content to the following two cases: searching for files containing symbolic representations of music; and searching for audio files containing music, instrumental sounds and sound effects.

Content based processing

Generally, multimedia content-based information retrieval systems do not perform any search operations directly on the media. They typically work with a set of relevant features which have previously been extracted from the media. In some cases, where there are no satisfactory algorithms to automate the feature extraction task, these can even be manually annotated. These features constitute to some extent a simplified model of the actual content and can be for example used to build an index for improved retrieval performance.

Even in the more restricted field of content-based retrieval of music, there is no universal set of features that will always perform well. Picking the right features for a particular search application can make a big difference and is often the key to success.

In the structural area however, an this could be extended to time-based media in general, a series of common guidelines that can be considered, and which are related to hierarchical representations of the data.

At the lowest level, a signal processing approach can be used to extract temporal and spectral parameters from the audio media on a frame by frame basis. These features are then used to characterize or summarize that particular frame. These parameters are frequently also called low level descriptors. The variability of one or more specific low level descriptors can then be used to infer the low level musical structure of the audio. For example, a sharp change in the fundamental frequency of the frames, may indicate a note change within that audio.

This leads to the concept of the segment, which brings us a level higher in the hierarchy, and which can be considered as a grouping of a certain number of audio frames that have a common higher level meaning. An advanced model should consider the possibility to describe hierarchies of segments, and at each level, there must be the possibility of defining and populating new descriptors. In our example, we can consider a descriptor for describing the note, and as we move up in the descriptor hierarchy, we will be able to define descriptors with a much higher semantic meaning, such as for example the tonality or even the composition style.

Obviously this model must allow for overlapping hierarchies of segments, where each of the hierarchies could have a different structural or semantic meaning.

A further dimension is added, if we have to take into account that this segmentation process is not always unambiguous; i.e. different systems (governed by different segmentation algorithms) can produce different segmentation results on a given musical content while trying to describe a particular feature. A good data model for information retrieval should support multiple versions of descriptions of a particular feature. This will also clearly be a requirement if the system is multi-user and has to support annotations of the content by different users.

The segmentation problem is not solved, and still a lot of ongoing research is trying to determine which are appropriate features for segmentation, how do these get combined in order to create higher level descriptors, and which are effective techniques for automating the segmentation process.

Retrieval versus Browsing

As in text-based information retrieval, an important differentiation must be made between the concepts of retrieval and browsing. In the first case the user is looking for a particular subset of sound objects in the collection. The concepts of precision (how many of the retrieved objects are relevant) and recall (how many of the relevant objects are retrieved) are adequate measures for evaluating the performance. In the second case, the focus is on discovery of new objects by following particular relations, analogies or rules. This relationships could be through any of the features which the model can have (same author, same title, same genre,…), or by combination of features.

The impact of these concepts on the data model is that it should support the notion of classes (in case of retrieval) or clusters (in case of browsing), and the assignment of the musical objects to one or more of such groupings. This allows for the possibility that a user can retrieve a desired musical object by picking it directly from any of the classes it belongs to. Classes can also constitute hierarchies, and if a musical object belongs to a given class, then naturally it should also belong to its parent class. As an example, we can consider a monophonic sound and a class hierarchy to describe the source of that sound. A first level could distinguish between instrument sounds, and sound effects. A possible second level could be to differentiate instrumental sounds in aerophone, idiophone, membranophone and chordophone; and so forth. A violin could be then classified into the chordophones class, and as a consequence it should automatically also belong to the instrumental sounds class.

Furthermore, this class assignments or belongings raise the need that the model supports the concept of references. This will have also impact in the design of a system that implements references as it will have to preserve the referential integrity. Alternative options could be to prevent removal of classes when there exist belongings to that class, or to automatically cascade the deletion of those belongings when a particular class is removed.

In addition, it should be noted that overlapping class hierarchies are required since a one single musical object could simultaneously belong to many classes. In our instrumental sound example, we already described a possible class hierarchy which could be used to classify sounds by sound source. In parallel a further class hierarchy could classify the main instrumental playing modes, and yet a further class hierarchy could exist to classify the timbre properties of the sounds.

Finally, and particularly for the browsing, the data model should support relations from one object to another. Analogous to the concept of anchors in hypermedia documents, users may be interested in linking musical objects to each other and a user in a browsing situation should be able to follow these relationships, which could be directed or not.

Advanced content based retrieval features

In [8], Aigrain presents an overview on new applications of content based processing of music. One of the most popular of the more advanced content based retrieval applications is the search by similarity, in which a reference object is used as the query.

Many variants of this problem exist and use different features in order to retrieve different similarities. A common approach is to search for timbre similarity in collections of instrumental sounds; i.e. search for instruments that sound similar [9], although timber-based similarities have also been used on collections of popular music titles.

In searches for melodic similarity [10] a reference melody is used as the query, and the system must find any titles or sequences that contain either the complete reference melody or fragments of it. Many factors such as the proper alignment of the reference melody, missing notes, rhythmical variations or variations in the length of the notes, and finally musical variations, such as transpositions of the melodies, make this problem especially complex and therefore attractive, as a good system may have to use a lot of "music theory intelligence". A particularly popular version of the search by melodic similarity is the "query by humming", where the reference melody is extracted from a hummed signal. This also introduces problem related to the quality of the reference pattern.

Searches by similarity do usually not constitute a problem in terms of database model, since they can be reduced to the problem of evaluating a distance function on a set of relevant features. There are however other types of database problems related to these types of searches, typically related to the low scalability of such systems.

In order to do an exhaustive search by similarity, the reference object must theoretically be compared to all the objects in the database, by evaluating the distance function, and frequently the cost of computation of that distance is high, making such an approach not feasible when the database is large.

Nevertheless, if the distance function complies a series of mathematical properties as for example the triangular inequality, then an exhaustive computation is note necessary and special index structures can be used in order to minimize the number of computations required. [11] Presents a good overview of the indexing structures available.

Another novel area in content based retrieval of music is the browsing through summaries [12]. This case is automatically covered if multiple levels of a segment hierarchy are allowed, and therefore does not pose particular difficulties in terms of the modeling of the data.

Other considerations

There are other factors which are not inherently related to the problem of modeling for multimedia, but which have to be taken into account when designing a real system for music information retrieval. The most relevant consideration in this area is the consideration that many descriptors will contain textual labels, which may need to be presented in several user languages and more important searched in a consistent user language independent way.

3. MPEG7 as a data model

MPEG-7 has been published as a standard with the objective to provide a common interface for audiovisual content description in multimedia environments. This should allow that different MPEG-7 systems or modules can easily interoperate.

The standard is based on the notion of Descriptors (D) and Description Schemes (DS). The former represent a model for specific high or low level features that can be annotated for a given media object. The latter just represent a grouping of a series of Descriptors or further Description Schemes in a particular functional area. As a matter of example, the standard defines the Agent DS, which among others groups descriptors to define a person, such as the GivenName, FamilyName and Title.

The definition of the MPEG-7 standard relies on other standards of the MPEG family and very specially on the XML language and the XML-Schema standard. MPEG-7 itself is provided in the form of an extensible XML-Schema defining an object oriented type hierarchy which delivers a set of predefined descriptors grouped into its functional description schemes. We recall the properties of the XML language as an adequate means for interchanging structured self-described information, and the XML-Schema language as a powerful tool (which overcomes many limitations of the Document Type Definitions) for describing the validation rules of a given XML document.

In [13] Kosch argues that MPEG-7 should not compete, but rather be considered as a complement to data models employed in Multimedia Database Systems.

In the context of the CUIDADO project [14] [15], we have studied the possibility of using MPEG-7 directly as a data model for a music authoring and post production application with implements exhaustive content based retrieval and processing functionalities and which we have called the Sound Palette. In our analysis, which covers the problems described in the preceding section, we have found that MPEG-7 standard already delivers the definition of many of the descriptors needed for the application.

Thanks to the extensible features of the XML-Schema language it has not been too complicated to supplement the standard with a set of new descriptors for melodic and rhythm description not originally available in the standard, and which are of particular relevance for the Sound Palette application. This extension can be made in form of a new application specific XML-Schema document that imports the standard XML-Schema document and creates a series of specialized descriptors such as the Scale, the Meter, the Key, the MelodyContour in the form of extensions of types already supplied by the standard.

```
<Mpeg7
  xmlns="urn:mpeg:mpeg7:schema:2001"
  xmlns:cuidado="urn:cuidado:mpeg7ext"
  xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:cuidado:mpeg7ext
     ./cuidado-mpeg7-ext.xsd" >
 <Description xsi:type=
     "ClassificationSchemeDescriptionType">
  <ClassificationScheme
        uri="urn:cuidado:cs:samplecs"
        id="id_sample_cs">
   <Header
      xsi:type="cuidado:BasicDescriptorsType">
    <cuidado:Label>A CS example</cuidado:Label>
   </Header>
   <Term termID="best_sounds" id="id_22">
    <Name>The sounds I like most</Name>
   </Term>
   <Term termID="ok_sounds" id="id_33">
    <Name>Sounds that are just OK</Name>
   </Term>
   <Term termID="bad_sounds" id="id_44">
    <name>I don't like this sounds</Name>
   </Term>
  </ClassificationScheme>
 </Description>
</Mpeg7>
```

Figure 1. MPEG-7 compliant Classification Scheme

The figure shows an example description in XML, also in order to get a feeling for the MPEG-7 description syntax. In this case a new classification scheme is defined, containing three new terms that other descriptions may be able to reference. The root element is the Mpeg7 element, and at that level both the standard MPEG-7 namespace, as well as the namespace for the CUIDADO project’s specific extensions are defined.

4. Database implementation

One very relevant aspect about MPEG-7 that should be stressed is that the standard does not define how the search or the indexing has to be made on the described data. It does not even assume that the internal storage should be done in XML format.

The two questions that come to mind are: Is the XML format of the data also an adequate format for the storage and retrieval, and in that case is it viable to implement such a system on an XML database?

For the database implementation, an Oracle 9i Release2 object-relational DBMS was chosen since it has a component (XDB) that can work with XML in a very efficient native way, and the DBMS has an good framework for extensibility.

The architecture of the system is as follows: The MPEG-7 XML-Schema that contains the definition of the data structures and types for describing the audio content is registered with the database, and thus creates the hierarchy of native MPEG-7 object types within the database, and a corresponding hierarchy of internal database objects that can store content for those MPEG-7 object types. The mapping is done automatically by the XDB on the basis of the MPEG-7 XML-Schema.

On the current prototype system the actual audio content is not stored inside the database but on the file system. This does not affect any of the results, since all the queries for retrieval are performed exclusively on MPEG-7 metadata, and not on the audio files. One single database table stores the XML descriptions, which can be on as many rows as required. There is no need for having all the descriptor database in a single XML document. That table has a column of XMLTYPE data type and is mapped to the registered MPEG-7 XML-Schema through a namespace within the DBMS.

At insertion time, the incoming description in XML format is automatically parsed and stored distributed across all the MPEG-7 internal database objects. This allows for very efficient access time to the data, as it does not have to be parsed at query execution time and for each new run of the query, as would be the case if the content would have been stored in the original textual format in a large object (LOB) column.

Furthermore, it is important to note that the storage is made in the native data types of each of the objects; i.e. a number is stored as a number (and not as text) and can therefore be indexed by the DBMS as a number.

Another good feature is that the object oriented mapping approach, provides an efficient data compression, since all the XML tags, which can be a high percentage of the overall description's volume, are not stored since they are not needed because the internal object types already preserve the MPEG-7 structure of the descriptions.

Retrieval is in general efficient, since all the important elements can be easily indexed with available B-Tree like indexes. In theory, there should be a slight penalty in performance in the reconstruction process of the XML from the objects. In practice, this does not pose any problems, since most of the queries are not meant to retrieve the full description as it was inserted into the system, but just a particular value.

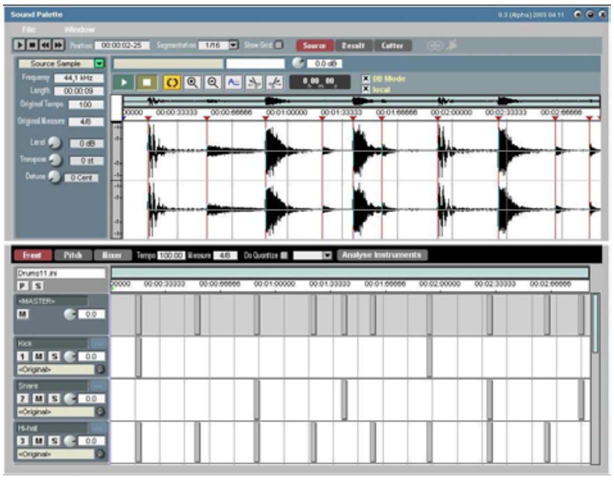
A structure has been created for providing special indexes that help in preserving the referential integrity of the descriptions. Although the DBMS allows to define foreign keys on the object types, we have opted to do a specific implementation of this in order to have a greater control over it, in addition, we can use this infrastructure as the basis for a search by similarity function that is currently being implemented and which should be available in the near future. This structure is maintained with a trigger on the table storing the descriptions, and automatically maintains our custom indexing structures.

The fact that the descriptor values can be easily indexed with B-Tree structures should give the production system a good scalability. Some limited but very promising testing has already been done. On a system running on a desktop PC, which had not specifically been tuned for the purpose, and with a data load of around 20.000 descriptions, the response times for an average query were under a second.

Although this particular database implementation has been made using only a specific subset of the MPEG-7 descriptors, those for audio content management, we believe that it can be considered generic from the point of view of meta data management and retrieval based on meta data information, since at database level it solely relies on XML and XML-Schema functionality, and therefore this approach should be able to be applied to the field of content-based modeling and retrieval of video applications as well, with very little adaptation efforts.

5. Application example

The Sound Palette is an application for content based processing and authoring of music and compatible with MPEG-7 standard descriptions of audio, which has been designed and implemented by the company Creamware in the context of the CUIDADO project. It is targeted for users who own large libraries of sounds and loops and offers novel ways to interact and work with audio.



Position 00:00:02-25
Segmentation 1/16
Frequency 44,1 kHz
Length 00:00:09
Original Tempo 100
Original Measure 4/8
Level 0 dB
Transpose 0 st
Detune 0 Cent
Tempo 100.00 Measure 4/8
Drums11.ini

Figure 2. Screenshot of the Sound Palette Application

Figure 2 shows a screenshot of the application’s content based features. A drum loop has been analyzed and segmented. This process has generated a number of descriptors which have been inserted into the database. The waveform for the drum loop is displayed in the upper region (2 channels since it is a stereo file), and the temporal segments are visually highlighted with vertical lines as separators.

In this case, the segmentation has also been made for the different instruments that build up the drum loop. This is visible in the lower region, which shows the two dimensions of the segmentation. The x-axis is still the time base, while the y-axis carries the different instruments encountered as a result from the segmentation. The beats corresponding to four different percussion instruments are visible on the lower region

This information, all stored in MPEG-7 format, can be then used to locate appropriate substitution sounds in the audio database. In the retrieval, many different MPEG-7 descriptors can be considered, such as related to physical properties (sampling rate, file format), related to ownership or rights holders, or more interesting parameters directly related to the actual content, such as the timbre, the energy, the pitch, etc.

In addition, the user has the choice to set tolerances for any of the numerical descriptors, in order to modify the retrieval accuracy of the system.

Finally, its interesting to mention, that Sound Palette allows that a user can organize the content collection with MPEG-7 classification schemes, which have been described in section 3. All this metadata information is stored in the database presented above in MPEG-7 compliant format, and can be used in order to either set further filters when searching for a particular content, or to directly retrieve the media from the virtual containers that the terms within the classification scheme constitute.

6. Conclusions

A system that is capable to perform content based manipulation, storage and retrieval of music, has been built and described in this paper.

The MPEG-7 has offered adequate tools to model the features used to describe the content in a satisfactory way from the user requirements perspective.

Furthermore, the popular XML based technologies that the MPEG-7 standard employs, have minimized the development effort by allowing the use of a widely available database management system capable of efficiently managing XML.

7. Acknowledgements

Part of this work has been supported by the European Commission, under the CUIDADO project (IST-1999-20194).

The author would also like to thank Oracle and Creamware for their support.

8. References

[1] W. Frakes and R. Baeza-Yates, Eds. *Information Retrieval: data structures and algorithms*, Prentice Hall, New Jersey, 1992

[2] M.Kobaiashi and K Takeda, “Information Retrieval on the Web”, *ACM Computing Surveys*, Vol. 32, No.2, June 2000

[3] J. Foote, “An Overview of Audio Information Retrieval”, *Multimedia Systems*, Vol. 7, No. 1, ACM Press/Springer-Verlag, January 1999

[4] F.I. Bashir, A.A. Khokhar, “Video Content Modeling: An Overview”, *Technical Report*, 09/2002.

[5] O.R. Zaïane, J. Han, Z. Li, J. Hou, “Mining multimedia data”, *Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research*, November 1998

[6] U.M. Fayyad, G. Piatetsky-Shapiro, P.Smyth and R Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996

[7] B. S. Manjunath, P. Salembier, T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*, Wiley, April 2002

[8] P. Aigrain, “New Applications of Content Processing of Music”, *Journal of New Music Research*, 28 (1999), No. 4, pp. 271-280

[9] G. Peeters, S. McAdams, P. Herrera, “Instrument Description in the Context of MPEG-7”, *Proceedings of the International Computer Music Conference 2000*, September 2000

[10] E. Gómez, A. Klapuri, B. Meudic, “Melody Description and Extraction in the Context of Music Content Processing” Journal of New Music Research Vol.32, 2003

[11] E. Chávez, G. Navarro, R. Baeza-Yates, J.L. Marroquín, “Searching in metric spaces”, *ACM Computing Surveys*, Volume 33 Issue 3, September 2001

[12] G. Peeters et al., “Toward Automatic Music Audio Summary Generation from Signal Analysis”, *Proceedings of the 3rd International Conference on Music Information Retrieval*, October 2002

[13] H. Kosch, “MPEG-7 and Multimedia Database Systems”, *ACM SIGMOD Record*, Vol 31, Issue 2, June 2002

[14] “Content-based Unified Interfaces and Descriptions for Audio/music Databases available Online” , www.cuidado.mu

[15] H. Vinet, P. Herrera, F. Pachet, “The CUIDADO Project”, *Proceedings of the 3rd International Conference on Music Information Retrieval*, October 2002

A System for Collaborative Music Composition over the Web

Sergi Jordà
Music Technology Group, Audiovisual Institute, Pompeu Fabra University Passeig de la Circumval·lació 8, 08003 Barcelona, Spain
sergi.jorda@iua.upf.es

Otto Wüst
Music Technology Group, Audiovisual Institute, Pompeu Fabra University Passeig de la Circumval·lació 8, 08003 Barcelona, Spain
otto.wust@iua.upf.es

Abstract

In this paper the authors propose a new architecture and new features for collaborative music composition. These design principles have been applied, without any loss of generality, to a system that has already been extensively tested on-line for the last three years, and which has allowed composers from around the world to participate in the collective creation of two important theatrical scores. They can constitute the basis for new approaches for collective composition on the Internet.

1. Introduction

Collective creation and the production of open and continuously evolving works are two of the most appealing artistic breakthroughs the Internet can offer to creators in general and to music composers in particular. The idea of musical computer networks is by no means original; earlier implementations (although on a local area scale) date back to the late 1970s with performances by the League of Automatic Music Composers [1]. However, twenty years later, collective music composition or improvisation on the net, is still at a bourgeoning state and sites and projects like *Res Rocket Surfer* [9], MIT's *Brain Opera* [8], William Duckworth's Internet based *Cathedral* [4] or the one we are presenting, *F@ust Music On-line (FMOL)* [7], can probably still be counted with the fingers.

FMOL is a system for real-time collaborative music composition on the web, started in 1997 and its third version is currently under development. Using a lightweight plug-in running on top of a web browser, FMOL allows users distributed all over the internet to work collectively on a single or on several musical pieces, sharing a common interface. It also permits new composers to modify and enlarge already existing pieces an endless number of times, while keeping at the same time the integrity of the original pieces.

2. History of a project

The FMOL project started during spring of 1997, when La Fura dels Baus, the Catalan theater group famous for its aggressive shows and performances that frequently involve audiences in an unpredictable manner, was beginning to prepare what would be its new show, *F@ust 3.0*, freely inspired in Goethe's work, and contacted us with an unusual demand: "*Given the important role symbolized by the Internet in our play, we would like part of its music to be composed by cybercomposers around the world*". Initially we did not have a clear idea of what we wanted; we just knew what we did not want: to allow people to compose tempered music on the keyboard and send us attached MIDI files via E-mail. Besides, although we felt that the project should have a fairly "popular" approach, and did not want to be therefore too demanding and restrictive about the participants' gear, we were not looking for a dull General MIDI sound, but for richer sounds and textures, that would match la Fura´s aesthetic approach and that could introduce newcomers into more experimental electronic music.

Real-time and mouse-driven software synthesis seemed therefore the natural solution for an affordable and at the same time experimental and sonically rich music creation environment. FMOL is consequently a net-based virtual synthesizer and graphical interface for real-time composition and synthesis, although, due to synchronization technical restrictions, we decided not to implement real-time communication (jamming) between its users[1]. Its collaborative approach follows instead a half-duplex communication paradigm, which enables composers to (a) listen to already existing pieces and to (b) optionally respond and collaborate on these pieces started by other composers, as well as to (c) start new pieces at any given time.

As we will detail, its architecture is based on a vertical-multitrack model (as opposed to a horizontal-*exquisite corpses* model, which would allow the pasting of sonic fragments one after the other). This allows each participant to start new compositions from scratch as well as overdubbing, modulating or processing any of the existing ones.

In should also be mentioned that FMOL experimental approach does not compete with other collaborative music projects and sites, like the popular *Rocket Network* [9], more focused toward bringing net communication facilities onto standard music production methods. FMOL emphasizes creativity and experimentation over production. We believe on one side, that the use of computers and new controller interfaces can bring a plethora of "new musics", and that, on the other side, the best way to understand and appreciate any discipline, whether artistic or not, and music is no exception, is by doing and being part of it. New and more *efficient* instruments can therefore bring new sophisticated music creation possibilities to non-trained musicians, or as Robert Rowe suggests, "let's develop *computer musicians* that do not just play back music *for* people, but become increasingly adept at making new and engaging music *with* people, at all levels of technical proficiency" [10]. The FMOL system is therefore not only constituted by a central server but it also includes proprietary synthesis engine and graphical user interfaces, providing a 100% web-based new collaborative composition environment.

3. Architecture

The first version of the system was built following a client server model. This allowed composers using the FMOL client software to log into a central server in order to download any of the pieces that were stored in a song tree structure. The composer was then able to work on some of the tracks of the piece with the standalone FMOL client, and send back the new version to the central server. Although the client server model has proven successful under specific circumstances, such as local, small sized productions, some of the major disadvantages of using this architecture were the installation process of the client, the redistribution and reinstallation of the client after a software upgrade or patch, and the inaccessibility of the music database to curious Internet surfers.

The second version of FMOL has been built according to a three-tier architecture model, which has proven to be one of the most efficient architectures for Internet computing. The server side hosts a database server, responsible for all the storage and retrieval functions. In the middle tier an application server is responsible for executing all the application logic.

The application server may be physically on the same machine as the database server or on a separate one, assuming that the network connection between both machines can support a high bandwidth and low latency. Furthermore, such configurations will allow a high degree of scalability. If a large number of simultaneous users need to be supported, several application server machines can be set up, and connections to the system can be handled by a load balancing service, which will distribute the petitions across the application servers.

Universal access to the system is guaranteed by the use of a thin client. Any *wintel* personal computer equipped with a soundcard and a standard web browser will suffice for running the FMOL plug-in.

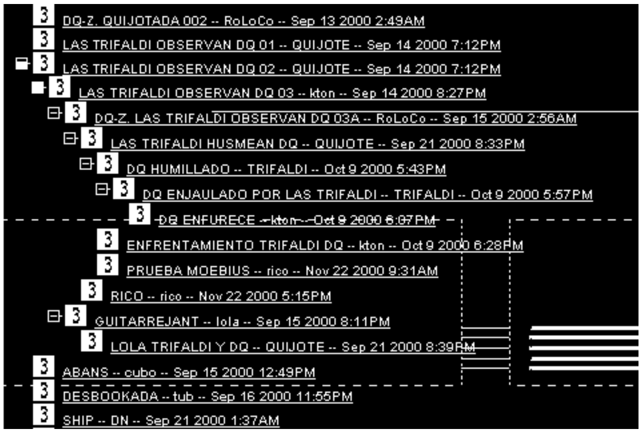
3.1. Database Tier

The FMOL system is based on a relational database. The main entity is the compositions table, which has a recursive relationship to itself, allowing for a tree like representation of the pieces. Each composition is represented by a node storing a pointer to a scorefile that holds the data for eight real-time synthesized audio tracks, which can be played by the FMOL plug-in's audio engine. A user can pick up any of the existing compositions, listen to it and work on it by overdubbing existing tracks or by adding a new track. The rework of a user is already considered to be a new composition. When the changes are posted to the database, the new composition is created as a child node of the one the user picked. This new child node will hold a pointer to the new scorefile. Therefore, the deeper a node is in the tree, the more revisions the piece will have had. It is in fact the tree structure itself which is implementing the version control. All nodes are public and the possibility that any user can modify any of the existing nodes enforces the collective composition approach.

Figure 1 shows a fragment of the compositions tree titled "3". Each line represents a node and displays the title of the composition, the author's alias and creation date. The amount of indentation reflects the depth or number of generations of the piece. In this case, several users have interacted to create up to 7 layers of

[1] This feature will be implemented in FMOL 3.0, which will be on-line on autumn 2001.

collaborative work, and some of the layers (i.e. 3, 4 and 5) have different siblings.



DQ-Z. QUIJOTADA 002 -- RoLoCo -- Sep 13 2000 2:49AM
LAS TRIFALDI OBSERVAN DQ 01 -- QUIJOTE -- Sep 14 2000 7:12PM
LAS TRIFALDI OBSERVAN DQ 02 -- QUIJOTE -- Sep 14 2000 7:12PM
LAS TRIFALDI OBSERVAN DQ 03 -- kton -- Sep 14 2000 8:27PM
DQ-Z. LAS TRIFALDI OBSERVAN DQ 03A -- RoLoCo -- Sep 15 2000 2:56AM
LAS TRIFALDI HUSMEAN DQ -- QUIJOTE -- Sep 21 2000 8:33PM
DQ HUMILLADO -- TRIFALDI -- Oct 9 2000 5:43PM
DQ ENJAULADO POR LAS TRIFALDI -- TRIFALDI -- Oct 9 2000 5:57PM
DQ ENFURECE -- kton -- Oct 9 2000 6:07PM
ENFRENTAMIENTO TRIFALDI DQ -- kton -- Oct 9 2000 6:28PM
PRUEBA MOEBIUS -- rico -- Nov 22 2000 9:31AM
RICO -- rico -- Nov 22 2000 5:15PM
GUITARREJANT -- lola -- Sep 15 2000 8:11PM
LOLA TRIFALDI Y DQ -- QUIJOTE -- Sep 21 2000 8:39PM
ABANS -- cubo -- Sep 15 2000 12:49PM
DESBOOKADA -- tub -- Sep 16 2000 11:55PM
SHIP -- DN -- Sep 21 2000 1:37AM

Figure 1. Screenshot showing a fragment of the compositions tree

A common problem in collaborative composition systems is intellectual property rights tracking. In our case, one of the design purposes has been to allow global access to musical collaboration. As a result many participating composers are casual Internet surfers, which makes this control even more difficult. The FMOL system implements a rights tracking option, which requires that a user is registered before allowing changes to the songs database. This control was used on the system's first implementation, in 1998, after an agreement with the Spanish authors' association, SGAE, who sponsored the project and facilitated all the registration proceedings even for non-associate authors. It has not been used, however, in the 2000 implementation.

Users are also allowed to vote on the quality of any composition. This information is stored in the database and can be used in various ways. First, the overall acceptance of a piece can be determined by the total number of votes scored, and this has proven helpful as an objective measure for the administrators of the system when some compositions have to be selected. Furthermore, we can take advantage from all the votes made by a single user, incorporating this information into the user's profile, which is useful for all the advanced query features explained in section 4.

3.2. Middle Tier

The middle tier hosts the application server and the web server. These software components are responsible for running most of the program logic of the FMOL system as well as serving the presentation layer to the web browser. We have opted for dynamic generation of the web pages using Java Server Pages (JSP) and for Java Beans to store the program logic. This includes all the web pages for user registration, profiling, voting, and most important, displaying the composition trees and managing the upload and download of compositions.

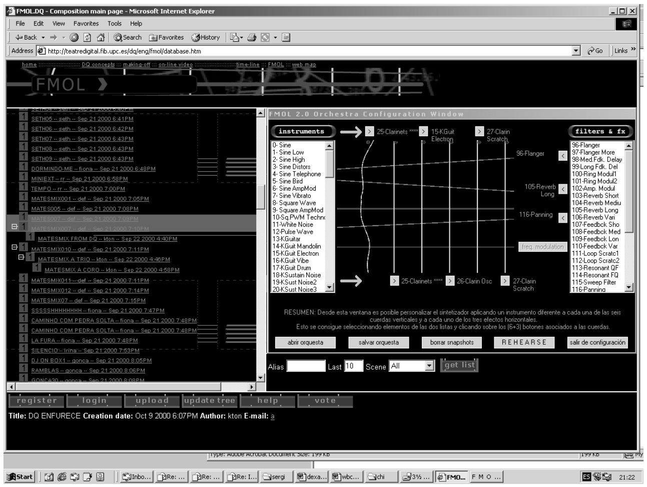
3.3 Thin client

The client tier is said to be thin because it only consists of a browser running a plug-in. The application logic is hosted mostly in the middle tier leaving the client layer only for the synthesis engine, the graphical interface and the presentation logic.

Despite the design objective of keeping the software running on the client to a minimum, there were both important esthetical and social reasons for including a specific proprietary synthesis engine, as one of the main objectives of the project was to approach experimental electronic music creation to newcomers and hobbyist musicians. In that sense, the FMOL composition and synthesis plug-in grants that everybody has access to compose, even surfers without any other audio software and no more hardware than a multimedia soundcard. This enforces an equal opportunity environment, while forcing at the same time, real-time composition and sound manipulation by means of innovative and intuitive graphical interfaces.

Although this three-tier architecture allows for different approaches which may be applied in the future without loosing any generality (as for instance the use of standard MIDI files or any other standard format which could be generated with currently available and generic software and without the need for a specific synthesis plug-in), the current synthesizer engine architecture and its graphical interfaces were in fact specially designed with this collaborative approach in mind.

The engine, written in C++ for the wintel platform, was meant to be a complete sound generation kernel flexible enough for real time synthesis and processing on a low-end machine (e.g. Pentium 200), that could be appealing and enriching for users with different skills and electronic music knowledge. The current version supports eight stereo real-time synthesized audio channels or tracks, each consisting of a generator (sine, square, Karplus-Strong, sample player, etc.) and three serial processors (filters, reverbs, resonators, ring-modulators, etc.) to be chosen by each composer between more than a hundred different synthesis methods or algorithms. Moreover, for each track (except for track 1) the called *generator* can in fact behave as a *parallel processo*r, which can take its input from the output of any of the lower channels (i.e. channel 5 can be configured to process channel 1, 2, 3 or 4). Each generator or processor possesses eight control parameters, four of which can be modulated by four independent low frequency oscillators (LFO), which makes a total of 128 LFOs (4 LFOs/algorithm * 4 algorithms/track * 8 tracks) that can be active simultaneously. The type of each LFO can also be dynamically configured (sinusoidal, square, triangular, saw tooth or random)[6]. All these parameters are updated at a fixed frame rate of 48 Hz.



FMOL.DQ - Composition main page - Microsoft Internet Explorer

Address http://teatredigital.fib.upc.es/dq/eng/fmol/database.htm

SETH05 -- seth -- Sep 21 2000 6:41PM
SETH06 -- seth -- Sep 21 2000 6:42PM
SETH07 -- seth -- Sep 21 2000 6:43PM
SETH08 -- seth -- Sep 21 2000 6:43PM
SETH09 -- seth -- Sep 21 2000 6:43PM
DORMINDO-ME -- fiona -- Sep 21 2000 6:48PM
MINIEXT -- rr -- Sep 21 2000 6:58PM
TEMPO -- rr -- Sep 21 2000 7:00PM
MATESMIX001 -- def -- Sep 21 2000 7:05PM
MATES005 -- def -- Sep 21 2000 7:08PM
MATES007 -- def -- Sep 21 2000 7:09PM
MATESMIX007 -- def -- Sep 21 2000 7:10PM
MATESMIX FROM DQ -- kton -- Sep 22 2000 4:40PM
MATESMIX010 -- def -- Sep 21 2000 7:11PM
MATESMIX A TRIO -- kton -- Sep 22 2000 4:46PM
MATESMIX A CORO -- kton -- Sep 22 2000 4:58PM
MATESMIX011 -- def -- Sep 21 2000 7:11PM
MATESMIX012 -- def -- Sep 21 2000 7:14PM
MATESMIX07 -- def -- Sep 21 2000 7:15PM
SSSSSHHHHHHHH -- fiona -- Sep 21 2000 7:47PM
CAMINHO COM PEDRA SOLTA -- fiona -- Sep 21 2000 7:48PM
CAMINHO COM PEDRA SOLTA -- fiona -- Sep 21 2000 7:48PM
LA FURA -- fiona -- Sep 21 2000 7:48PM
SILENCIO -- Irina -- Sep 21 2000 7:53PM
DJ ON BOX1 -- gonca -- Sep 21 2000 8:05PM
RAMBLAS -- gonca -- Sep 21 2000 8:06PM
GONCA30 -- gonca -- Sep 21 2000 8:08PM

FMOL 2.0 Orchestra Configuration Window

RESUMEN: Desde esta ventana es posible personalizar el sintetizador aplicando un instrumento diferente a cada una de las seis cuerdas verticales y a cada uno de los tres efectos horizontales.
Esto se consigue seleccionando elementos de las dos listas y clicando sobre los (6+3) botones asociados a las cuerdas.

Alias Last 10 Scene All

Title: DQ ENFURECE **Creation date:** Oct 9 2000 6:07PM **Author:** kton **E-mail:** a

Figure 2. Screenshot showing a part of the tree database (left frame) and the FMOL plug-in configuration window (right frame)

This architecture allows any composer not only to add new sound layers to previous compositions, but also to apply further processing to any of the composition's existing tracks, modulating or distorting what other composers did, in unpredictable manners. That way, a musical idea brought by one composer can grow and evolve in many different directions unexpected by its original creator.

4 Collaborative approach new features

Current work consists in refining and enhancing the server side features of the system. The overall objective is to use the stored information to enforce the collective composition approach. The main techniques applied are user profiling and content-based retrieval of the compositions. This enables the FMOL system to automatically propose compositions to the users for them to work on, according to their respective preferences and inferred taste and interests.

4.1 FMOL file format and transformation into XML

Compositions in the FMOL system are stored as scorefiles consisting of time stamped commands for the real-time synthesis engine. Each scorefile has a fixed-length header and a variable part which stores the multiple tracks created by the multiple authors.

In order to do content based processing and analysis of the information, we have found adequate to have the information in XML format. This allows for easy parsing of the recorded attributes and events. It is even possible to store the XML file as a large object in the database, accessing and indexing the individual attributes. An FMOL-XML transformation component is currently under development.

4.2 User profiling

By means of user profiling [5], a system can gain knowledge about the preferences of a given user. The system can then take advantage of this information for various purposes, such as suggesting the most adequate partners for collaboration, or the most adequate musical pieces for participation in collective composition.

In FMOL, the user profile information is acquired in several ways. Through a preferences section, the user can actively enter subjective information, such as his interest in musical genre, favorite instruments, musical training and level of expertise, etc.. In addition, FMOL will monitor the user's behavior and interaction with the system. Through the compositions that a user chooses for collaboration and the votes he submits, FMOL can cluster the authors into virtual communities. Furthermore, for each of the pieces published by a user, FMOL will automatically extract and infer objective information about the composition, such as density of the notes, rhythm, melodic lines, orchestration, etc.. This profile information is stored as feature vectors that form an n-dimensional space. We are currently evaluating several of the existing techniques for performing similarity queries in such feature vector spaces.

The system is constantly being tuned towards the preferences of the users by taking knowledge of their feedback. By using its profile information, it can propose a list of pieces for the composer to work on, according to his/her preferences. After working on a piece that has been suggested by the system, the author can evaluate the quality of the proposal, and this information will be stored in the system and taken into account in its next proposal.

4.3 Content based retrieval

Another new feature of the proposed architecture is the inclusion of content-based retrieval functions. Since the musical information of the pieces has been extracted into XML structured feature vectors queries can be performed in this feature space. One of the problems to consider is that most of the extracted features are not significant to the end composer. Previous work in the area of content-based retrieval of music has used the notion of melodic contours [2] which employs the melody profile extracted from a midi file. We are therefore currently trying to exploit the properties of the synthesis algorithms to perform a mapping between low and high level descriptors. This should allow end user queries by similarity using high-level criteria, such as the notion of similar instrumentation or similar playing modes.

5. Real-time users' interaction – Net-jamming

FMOL versions 1 and 2 discarded the implementation of real-time interaction between different users, mainly because of synchronization technical restrictions, but this important feature will be finally available in the new version, thus allowing several players to share a common environment (i.e. to improvise together).

Real-time asynchronous and multipoint messaging through the Internet poses serious timing and synchronicity problems. Different implementation decisions have to be made, from a peer-to-peer versus client-server architecture to an input versus state synchronism. Latency and synchronicity needs from a musical performance point of view in general and for FMOL in particular, do not differ too much from the ones needed for multi-user action games. Typical latency values for MIDI based Internet applications can range, considering standard 56K connections, between 500-1,000 ms, which is unacceptable for most musical styles and action games. However, FMOL's compositions are more timbrical than rhythmical and can therefore better tolerate the variable delays caused by the latency, in a similar fashion as the Gregorian chant dealt with the several seconds long reverberation times of the cathedrals.

The system we are building for this purpose is a real-time messaging server based on Phil Burk's *Transjam* [3] protocol. This server resides on the middle tier and implements the FMOL session manager. A web-based interface monitors the active sessions and the participants in each of them, allowing a user to create new sessions or to enter any of the currently open ones (given that the maximum number of clients/session has not been reached).

At every frame (i.e. 48 times per second), each client sends the generated events to the server. FMOL typical data rates range from 60-180 bytes/client.second. The server, running at the same frame rate (but no necessarily synchronized) redistributes all the received messages to all the clients. We consider a priority that each user can control his/her interface with no appreciable latency. Therefore the client's own messages are treated directly before they are processed by the server, and when the server sends messages to this client, this client's messages are filtered out in order to avoid feedback.

This mechanism leads however to an unavoidable compromise, implying that every client of a session will be listening to a slightly different version of the same piece. We are currently evaluating the possibility of embedding time-stamps into the messages to allow for periodic server side resynchronizations that would minimize these differences.

We must finally point out that this real-time interaction should not be seen as a replacement, but rather as a complement of the existing collaborative possibilities discussed in the previous sections.

6. Musical and Social Implications

Two FMOL versions have so far successfully been used by hundreds of Internet composers, as a virtual electronic music instrument for the collective composition of two scores for the la Fura dels Baus, including the play *F@ust 3.0* and fragments of the multimedia opera *Don Quijote en Barcelona*, premiered at the Gran Teatre del Liceu of Barcelona in October 2000.

From January to April 1998, the FMOL first Internet-database received more than 1,100 brief pieces by around 100 composers, some of whom connected nightly and spent several hours a week creating music. One of our main goals (i.e. to conceive a musical system which could be attractive to both trained and untrained electronic musicians) was fully attained. We know now that several of the participants had no prior contact with experimental electronic music and a few were even composing for the first time, but all of them took it, however, as a rather serious game, and the final quality level of the contributions was impressive. After a difficult selection process (only 50 short pieces could be chosen and included on the show's soundtrack), and considering that a great number of interesting compositions had to be left aside, we decided some months later to produce a collective CD with a mixture of old and new compositions.

A new web with a new version of the software has been back on-line during September 2000 for La Fura´s new show, the opera DQ, premiered last October at the Gran

Teatre del Liceu in Barcelona[2]. During one month, more than 600 compositions have been submitted, and the selected ones constitute now the electroacoustic parts of an otherwise orchestral score. A third version with new features will be available in autumn 2001.

7. Conclusions

This paper has presented a new approach to architecting and building a system for collaborative music composition. By successfully using these design principles in a real system implementation, FMOL, we have proved the viability of our proposals. Furthermore, we propose new ideas for collective composition environments. These are serving as a basis for current and future work.

8. References

[1] Bischoff, J., Gold, R., and Horton, J. Music for an Interactive Network of Computers. *Computer Music Journal, Vol. 2, No.3, 24-29, 1978.*

[2] S. Blackburn, D. DeRoure, “A Tool for Content Based Navigation of Music”, *ACM Multimedia 98, Electronic Proceedings.*

[3] Burk, P., “Jammin’ on the web – a new Client/Server Architecture for Multi-User Musical Performance”, *Proceedings of the International Computer Music Conference 2000.*

[4] William Duckworth's Internet based Cathedral piece: http://www.monroestreet.com/Cathedral/main.html

[5] T. Fawcett, F.J. Provost, “Combining Data Mining and Machine Learning for Effective User Profiling”, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press, 1996, pp. 8-13*

[6] Jordà, S. A graphical and net oriented approach to interactive sonic composition and real-time synthesis for low cost computer systems. *Digital Audio Effects Workshop Proceedings, 1998.*

[7] Jordà, S. Faust music On Line: An Approach to Real-Time Collective Composition on the Internet. *Leonardo Music Journal, Vol. 9, 5-12, 1999.*

[8] Tod Machover and the M.I.T. Media Laboratory Brain Opera: http://lethe.media.mit.edu/first-page.html

[9] Res Rocket Surfer site: http://www.rocketnetwork.com/

[10] Rowe, R. Interactive Music Systems – Machine Listening and Composing, The MIT Press, Mas, 1992. p. 26

[2] Visit the web, download the software or learn more about the DQ-FMOL project at http://teatredigital.fib.upc.es/dq