Using natural language input and audio analysis for a human-oriented MIR system

Stephan Baumann
German Research Center for AI
Erwin Schrödinger Straße
67663 Kaiserslautern
Stephan.Baumann@sonicson.com

Andreas Klüter
sonicson GmbH
Luxemburger Str. 3
67657 Kaiserslautern
Andreas.Klueter@sonicson.com

Marie Norlien
German Research Center for AI
Stuhlsatzenhausweg 3
66123 Saarbrücken
Marie.Norlien@dfki.de

ABSTRACT

In this paper we will present a MIR (Music Information Retrieval) system using natural language as input for human-oriented queries to large-scale music collections, applicable in web databases, cd-chargers for cars, or mobile services. The outlined system is a full-fledged architecture combining state-of-the-art approaches from the fields of natural language understanding including phonetic matching, automatic analysis of audio data for the purpose of meta tag construction, content-based classification of audio and music ontologies as a backbone for the representation of musical knowledge. On top of this architecture different prototypes for industrial applications are described including first results of real-life field tests. This work has been performed at the German Research Center for AI and the authors spin-off company, the sonicson GmbH.

1. INTRODUCTION

The digital distribution of music is one of the most attracting and challenging topics for musicians and computer scientists these days. In despite of the ongoing legal debates we find a lot of potential for convenient man-machine-interfaces to music on the technical side. The focus of this paper are applications for so-called *music information retrieval* (MIR).

Our goal is the provision of an underlying system architecture giving as much flexibility as needed to build powerful applications as customized instances of a multi-component approach. In this way our system approach can be applied for the main application areas in this field:

- Digital music distribution (e.g. Napster, Musicnet, Pressplay)
- Online retailers (e.g. Amazon, JPC)
- Music information services (e.g. Musicline, All-music-guide)
- Retrieval and playlist generation for hardware music devices (e.g. portable MP3 players, CD charger for cars)
- Mobile services (future UMTS services, PDAs)

For all of these applications our approach meet some goals with respect to convenient usability and a minimum amount of manual indexing of underlying large-scale musical data:

- Human-oriented interface paradigm: utilizing the expressive power and simplicity of natural language requests
- Fuzzy interpretation of misspellings: phonetic matching
- Beginner and expert handling: expressiveness, transparent interfering from symbolic to sub symbolic concepts
- Uniform feature handling: administrative metatags, lyrics, musical and audio features
- Automatic acquisition of features: using standards, web crawling, automatic audio and text analysis
- Retrieval, recommendations: uniform classification in a VSM model (Vector Space Model)
- Usage of extendible common-sense knowledge about music: music ontology alignment

2. ARCHITECTURE

The functional elements of the MIR system BEAGLE are realized in the system design as independent components. Figure 1 shows the interrelationship of *classifier*, *audio analysis* and natural language interpretation including the phonetic fuzzy match (*NLP Interface)*.

The components *DB manager* and *customer interface* implement flexible interfaces which simplify the integration work in existing customer environments (eCommerce solutions, databases, etc.). The service module *scheduler* performs the executive control of the whole system as well as the quality of service management for simplified scalability of the MIR system. The music data of the customers are accessible in the *music-DB*. *Name-DB + corpus* determine the country language and the natural language processing. They are interwoven with the language-independent musical concepts specified in the ontology.

Simplified, a possible query *"like lucky star by Madonna, but faster"* is processed as follows: Initiated by the *Customer Environment and Service* and handed over by the *Customer Interface,* any query is passed to the *Scheduler*. An instance of the *NLP-Interface* identifies the semantic parts of the input sentence: (a) DB-ID: madonna/lucky_star, (b) result should be similar to that and (c) but faster. Since a similarity fingerprint of

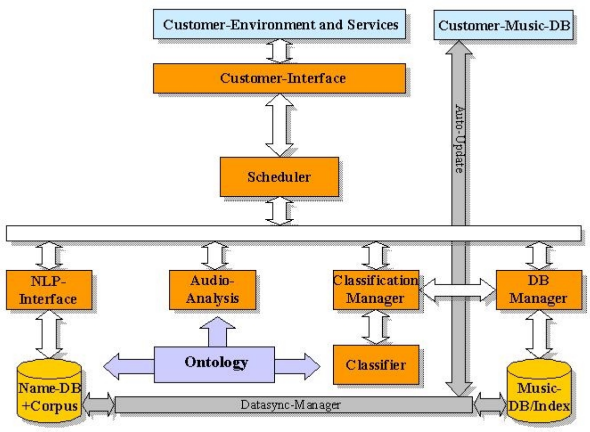

Figure 1. System Architecture.

"lucky star" was already extracted in a preprocessing step, the *Audio Analysis* component is not employed in this example. Based on its' internal know-how and information from the *DB Manager*, the *Classification Manager* extracts a set of matching titles. To finish the query, the results are passed back to the user via *Scheduler* and *Customer Interface*.

In the following subchapters the individual components are highlighted.

2.1 Ontological backbone

Answering real life questions requires real life knowledge in the music domain to be used within the MIR. The sonicson system utilizes a ontology about the domain of music for this purpose.

"An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary" [1].

In our application scenario we noted as terms the concepts of required know-how in the music domain. The relations consist of several types, namely is-a and part-of relations are used quite often. Is-a-relations are used to indicate specializations of concepts (e.g. *acid jazz* is-a *jazz*, *organ* is-a *keyboard*) while part-of-relations denote required parts (e.g. *track* part-of *compilation*, *member* part-of *band*). In this way we were able to set up an initial ontology for further refinement and alignment with other existing ontologies.

We use the Protégé 2000 tool for convenient design of ontologies. This tool and the underlying approach have been developed at Stanford university, mainly driven by needs in medical departments [2]. Protégé 2000 is a cross-platform tool under widespread usage in different areas. It is mainly its flexible plug-in-architecture and the powerful import and export of standard formats such as XML and RDF(S) making it the tool of choice when it comes to ontology editors. Beside the definition of classes, automatically generated form editors allow the definition of the instances.

Especially the options to import other know-how domains via importing RDF(S) files and the open plug-in architecture have been the major reasons to choose PROTÉGÉ 2000. This is important with respect to the ongoing developments of ontologies at different sites for different applications and in different

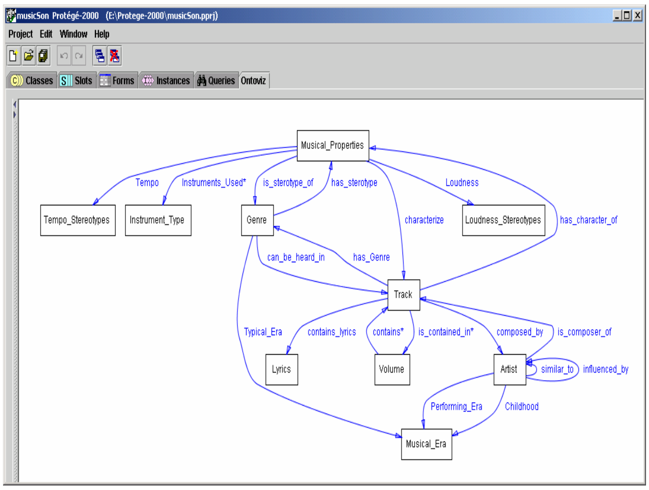

Figure 2. Excerpt of ontology.

formats. These problems are described thoroughly in a recent publication of Pachet [3]. A further difficult question is how to handle the acquisition of instances which represent beside the generic concepts the actual state of the "musical world". This will be an ongoing and dynamic process since "music never stops": new artists, new releases, the latest trends in genres, etc. pop up at different corners of the world and have to be assembled.

2.2 Grounding by music database

The MIR system accesses the musical data from an underlying database. In our first prototype we ripped a private CD collection to MP3 format at 128kbps. The scope of this simple database is about 1000 tracks covering 60 artists and approx. 50 different genres. The administrative information about artist, title, volume have been gathered by usage of the CDDB (Gracenote, [4]). The genre tags have been set manually.

For other applications, e.g. an environment with CD chargers for cars, the database scheme is the same, but the links to the audio data allow for different formats such as PCM linear 44.1 kHz / 16 bit.

2.3 Intelligent agents for information acquisition and processing

A bunch of different components is used to bridge the gaps between user's utterances and intended needs to the concepts in the ontology and finally to access the musical data in the database.

2.3.1 Usage of ID3 metadata

The first try to receive meta information based on MP3 content is to take a look at the ID3 tags [5]. Unfortunately, in real life data quality is insufficient for automatic processing. This is because meta information found in ID3 tags mainly come from databases like Gracenote [4] or the FreeDB project [6]. Those databases are generated by a large community of volunteers and their input – though very useful in many applications – is not quality assured. While the inconsistencies in artist, title and volume tags can be removed automatically [7], the genre information remains useless for automatic processing since the tagging is far too heterogeneous – if this ID3 tag is filled at all.

The upcoming MPEG-7 specification is expected to be a great step towards the semantic web, while on the other hand it is not clear how large scale high quality data will be provided to utilize the advantages that the MPEG-7 standard offers.

2.3.2 Automatic audio analysis

The automatic audio analysis recognizes properties such as loud/quiet, fast/slow etc, as well as more sophisticated features for the determination of similarity.

Here, sonicson uses a psychoacoustic model: only the characteristics in the music which are clearly recognized by the human ear are used. Simultaneously, the size of the fingerprint which is compiled is kept low (about 300 KB).

For the extraction of basic features such as loudness and psychoacoustic features, we used the approaches of Pfeiffer which have been compiled in a toolset under GPL license, available at CSIRO [8]. For the purpose of tempo analysis we developed our own approach which consists of techniques described in [9]. The extracted features are stored as a feature vector in the database.

2.3.3 Natural language interpretation

Approaches to processing natural language lie between the two extremes *key word processing* (= disregard for word relations and context) and *complete understanding*. Both are not applicable for pragmatic processing of natural language music queries.

The approach of *example-based processing with partial abstraction* [10] is especially suited for music search requests (limited domain, high speed requirements) and offers an optimum trade-off between:

- High processing speed: Intensive abstractions are only carried out where this is unavoidable (artists names, genres, special natural language phenomena, etc.).
- Simple maintenance and servicing: Code and corpus are clearly separated which means that further extensions (more languages, new trends) have no effect on the implementation. Additionally, the corpus can be compiled and extended by personnel who have been trained on-the-job but who are not necessarily highly qualified.
- Good-natured reaction to off-scope requests (requests whose complexity reach the limits of the machine processing of natural language).

2.3.4 Correction of phonetic misspellings

BEAGLE is not confused by typing errors. Additionally, the system is able to connect artist names which sound similar to each other, i.e. it is still able to produce results when there is phonetic similarity (such as e.g. „fil collins“ vs. „phil collins“).

Many general purpose sequence distance methods have been investigated [11][12][13]. The phonetic fuzzy match used by our system is based on an outstanding doctorate thesis on this subject. In contrast to other methods of non-exact search (such as Levensthein [14] or Soundex [15]) our method is optimized for application in the music field. Some of the requirements that were critical to be solved within this scope are:

- two possible sources of misspellings: (a) phonetic errors and (b) typos have to be handled in parallel
- automatic conversion of database entries into phonetic representations: artists from different countries require modified phoneme (rule) sets
- scalability of the phonetic fuzzy match for up to 2-3 million database entries: smart clustering strategies

2.3.5 Content-based retrieval and classificaton

Large music databases should provide efficient and easy access. So-called content-based music retrieval and classification is a well-known research topic being explored by different authors in the past [16]. Still the most challenging task is the selection of the most appropriate low-level features in combination with a well-suited similarity measure, resp. classification approach.

Our architecture may be used as a testbed for different classifiers being coordinated or voted by a classification manager which synchronizes the different results. Actually we are working with a standard Nearest Neighbor (NN) classifier to deliver cross-genre recommendations for music “sounding” similar. The basic audio features are MFCCs, bpm and loudness. A temporal clustering is performed in combination with the NN classifier.

Additionally we carried out first experiments using support vector machines (SVM) [17] to classify new titles according to pre-trained classes by genre. In our experience we found it very difficult to create a sufficient amount of groundtruth data for training. We are still working on this topic in an experimental stage.

2.3.6 Exploring lyrics with document retrieval

We started some experiments concerning the similarity of lyrics and the implications for the perception of music similarity. Here we use state-of-the-art document retrieval and classification approaches which have been recently commercialized and successfully adopted to real-world problems [18][19][20]. We used both, an API to the MindAccess API of Insiders AG as well as the text classification workbench and its submodules developed at our institute [21].

These tools allow for different functionalities. A query in the boolean retrieval model consists of a boolean combination of tests on the occurrence of specific words. For instance, the query *(hate or love) and girls* tests whether a document contains one of the words *hate* or *love* as well as the word *girls*. In this way a lot of the users questions about the content of a song can be handled.

To go beyond the boolean retrieval, additional functionality which we integrated is based on the vector space model (VSM). In this model, lyrics as well as queries are represented as vectors. The dimension of the vectors indicate specific terms, the value of a vectors component indicates the number of times the respective term occurs in the lyrics/query to be represented. Standard document retrieval based on queries in the VSM is done by defining a similarity measure between vectors. The most frequently used measure here is the cosine-measure, which computes the angle between two vectors. Having a vector representing the query, the documents corresponding to the most similar document vectors are returned as answer documents. In this way we realize the computation of similarity among lyrics. Since queries and lyrics in the VSM are represented as vectors, also the similarity between vectors representing just lyrics can be computed. Roughly spoken, those lyrics, which share many important words, will have a high similarity.

This tool has only been tested in isolation and has to be integrated into our system in the future. Our initial tests with students interested in music showed how people like to play around with such a tool. This seems to be an indicator for its importance to create “stickiness” in future digital music services.

2.4 User interfaces

2.4.1 Google style

Figure 3. “googling” for music.

A very intuitive interface to a web-based music distribution service consists of a form with one single entry field. No additional buttons for a contextual refinement of the query are required, since the entire processing of the query allows for extraction of the required context. In order to stay very simple and therefore user-friendly we designed our interface to BEAGLE in a “google”-like style.

2.4.2 Go mobile

The future developments in the mobile devices and application areas are demanding for specific interface design. Without keyboard and mouse entry a natural-language input may be the paradigm of choice. In this case an additional speech recognizer would translate the speech input to an intermediate ASCII representation which serves as input to our approach.

Actually we are implementing such a scenario in cooperation with the research group for ubiquitous computing at FH Trier [22]. Our platform is a Compaq Ipaq with a speech recognizer and a wireless LAN connection to our server. Future work will report about the experiences in this scenario.

Figure 4. PDA Interface of the MIR System.

3. REAL WORLD EXPERIENCE

The described system approach has been developed one year ago. Right from the start we coped with real-life test scenarios and applications. The first system implementation started with a German natural language interface. We made a collection of unrestricted typical queries covering test persons in the full range from beginner to expert. From these examples we compiled the first German corpus for the NLP component. A few month later we did the same with US natives, again covering a range of amateurs to music professionals. The results have been very similar to our German experience. A further evaluation of our approach has been conducted for a web-based German music information system. We will make some concluding remarks about these experiences in the following subchapters.

3.1 Evaluation of the expressiveness

In this section we show examples of typical request categories on the MIR system. The categories were initially developed by the Evaluation Center for Language Technology Systems at the German Research Center for Artificial Intelligence (DFKI, see also section 3.3) during the first Evaluation stage that started in July 2001. In total, about 1500 requests were collected and analyzed.

3.1.1 Requests for artist

These queries are all regarding artists. It makes use of only the artist's name.

'Phil Collins', 'don henley', 'garth brooks', 'james taylor', 'Madonna', 'britney spears', 'britney', 'george micheal', 'Michael Jackeson', 'Jackson', 'Lionel Riche', 'celene dion', 'dion', 'britney', 'britney spears', 'Lionel Riche', 'Janet Jackson', 'Harry Connick Jr.', 'Harry Connick Jr.', 'Vanessa Mae'

Here it is important to notice that capital letters are not necessarily used and sometimes the complete name is not present. But without taking capital letter problem into account, it showed up that for over 35% of the requests, the query did not directly match the corresponding database entry and would thus have come out with no result using a simple database lookup.

3.1.2 Wrong spelling

These queries are all misspellings of artist's names or groups/bands. Here it is important to notice we included a lack of space between words as a misspelling.

'britnes', 'maonna', 'Backtreet Boys', 'BackStreet Boys', 'BackStreet Boys', 'Madona', 'Micheal Jackson' , 'Celine dion'

Nearly 90% of the misspelled requests could be satisfied utilizing our phonetic fuzzy match approach.

3.1.3 Songs

These types of queries appear when the user knows what exactly he/she wants to find. Again, here we have the use of full sentence queries and capital letters (used or not used).

'best of my love', 'crush by dave matthews', 'in your eyes by peter gabriel', 'George Michael I Want Your Sex', 'I would like to find the song Crazy for you.', 'Do you have the Lion King soundtrack' 'soundtrack from stand by me', 'janet jackson and go deep', 'fleetwood mac and dance', 'homeward bound' 'Mariah Caray Butterfly', 'Get It On Tonight', 'Eagles Live'

Even if only looking for a specific song, users noticeable expected enhanced functionality of the interface: Combinations of artist and title, real world knowledge about soundtracks and other concepts. All of that regularly mixed with misspellings.

3.1.4 Questions

These queries seem to be trying to find out general information versus specific music by artist's and/or groups/bands.

'what cd's did madonna made between 1985 and 1995'

'What CDs did Toto make between 1975 and 1980?'

'Latest CDs'

'songs from the Eurythmics'

'miles Davis first album'

'What is the most popular song today in pop music?'

'What albums are available by Madonna'

'Do you have music by Madonna'

'I would like music by Phil Collins'

'Do you have a list of musicals'

'Do you have movie soundtracks

'Michael Jackson's first album'

'music from the eagles 1971 to 1990'

'music by the dave matthews band'

'number ones by simon and garfunkel'

'all songs by rush'

'U2 latest album'

'new album'

'new releases'

This is the type of query preferred by users that want to draw a benefit of the knowledge stored in a large scale music database. For example, they may know there was a popular song in the late 80's by Madonna that they liked, but they cannot remember the specific title. Or the user might just prefer a more natural language of full sentences for a user-friendly interface.

3.1.5 Music Categories

'Jazz artists', 'rap, hip-hop', 'blues, jazz', 'Christmas tunes' 'Kitaro', 'Jazz Music', 'jazz', 'acid jazz', 'easy listening', 'easy rock', 'African music', 'Contemporary Violin', 'Modern violin', 'Foreign Music', 'new age', 'Opera', 'Classical'

These queries are mostly like used by the inexperienced user looking for music in general, but by category. Additionally, users that are interested in finding something new might use these queries. Or perhaps a friend had a music category suggestion like you should try jazz music. It is important to note that the genre request may be combined with further constraints like 'artists' in 'jazz artists' or 'tunes' in 'Christmas tunes' which the NLP Interface can extract and influence the search and the result presentation.

3.2 Evaluation of phonetic matching

We checked our phonetic matching approach on the real life web log files of a German music information system. The following show some of the "highlights" regarding user queries in the domain of artist names:

Query	*DB entry*	*match*
status que	*Status Quo*	✔
golgen earing	*Golden Earring*	✔
jil caplan	*Kaplan,Gilbert*	✔
Eva Kessidy	*Cassidy,Eva*	✔
xseviar naido	*Naidoo,Xavier*	✔
deepest purple	*Deep Purple*	✔
Fisher Set	*Fischer Z*	✔
Novospaski Chor	*Novo Spassky Chor*	✔
daido	*Dido*	✔
four none blondes	*4 Non Blondes*	✔
Browne, Crosby, Stills, Nash, Young	*Crosby,Stills,Nash & Young*	✔
Matchbox twenty	*Matchbox 20*	✔
ACDC	*Ac/Dc*	✔

Table 1. Results of phonetic match.

3.3 User Testing of the Example-Enhanced Query Interface

The sophisticated natural language processing of BEAGLE does not in itself guarantee that users will immediately be successful in finding the music of their choice. A precondition is that users will have some idea of the types of queries that they can formulate. Informal tests have shown that some users restrict themselves to the sort of input that they are accustomed to making with web search engines: For example, they may simply enter names of artists; since they receive some useful results, they may never notice the that they are missing out on the opportunity to formulate much more flexible queries.

In cooperation with the recently founded Evaluation Center for Language Technology Systems at DFKI [1], we are exploring ways of presenting users with a few carefully chosen example queries that will quickly suggest to the user the kinds of queries that can be entered.

In an informal study with several visitors at the CeBIT 2002 trade fair in Hannover, we presented each user with the basic BEAGLE interface augmented with some example queries. Since the design of the presentation of such examples requires careful attention to detail, we used a remote eye tracker to record each user's eye movements as they formulated each of several queries and examined the results returned.

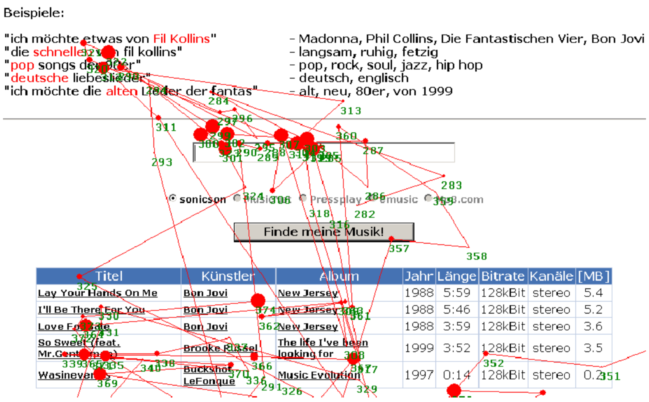

Figure 5. Eye movements of a user posing a query to an early version of the example-enhanced interface. ***(Each dot in the figure represents one fixation by the user, the size of the dot reflecting the length of the fixation. Numbers indicate the order of the fixations.).***

Figure 5 shows how one user looked at an early version of the example-enhanced interface. The example section comprised two parts: five example queries (on the left) and five corresponding sets of concepts (on the right) that could be used in place of the highlighted words in the example queries. It can be seen that this user did not even look at the right-hand column of examples. A retrospective interview with him revealed that he assumed that the right-hand column would contain just more examples like those on the left. Moreover, the distinction between examples and possible variations on them seemed to be too complex for users who are eager to enter their first query. This user also pointed out that two of the five examples did not demonstrate any *added value* of natural language queries relative to the simple entry of names. For example, in "I would like something by Fil Kollins", the words "I would like something by" do not add anything to the results that would be obtained by typing "Fil Kollins".

On the basis of similar results with other users, we designed the screen shown in Figure 6. The five short examples were intended to encourage users to exploit the power of the natural language query analyzer. The first example (which means roughly "Something snappier from 2002!") was printed in huge orange letters so that users would be sure to read at least this one example. As the fixations from one user (reproduced in the figure) show, the examples did indeed receive close attention. On the

[1]The Evaluation Center is part of a Competence Center for Language Technology Systems at the German Research Center for Artificial Intelligence (DFKI) that is being funded by the German Federal Ministry of Education and Research (BMB+F).

other hand, the one extremely salient example was ignored: The user stated afterward that he assumed that it must be a general heading or slogan, the kind of thing that his long experience with web browsing had taught him not to look at.

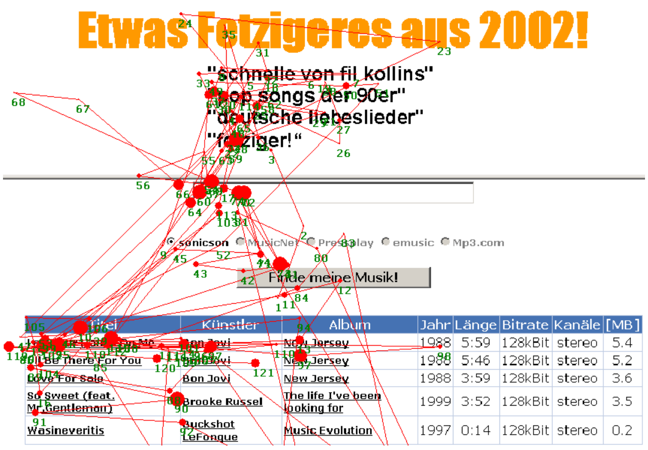

Figure 6. Eye movements of a different user posing a query to an improved version of the example-enhanced interface..

Although this study involves only a few users, it does illustrate several general points:

- Novel means for users to express their musical desires may not be exploited if users are not made aware of the possibilities available. In particular, habits that users have acquired with traditional query methods can interfere with effective use of novel methods.
- Query examples can effectively suggest appropriate forms of user input, but they must be designed and tested carefully, lest important aspects of the examples be ignored.
- Eye tracking can be used effectively even for quick iterative evaluation of query systems. Fixation plots such as those in Figures 5 and 6 can make flaws in a screen design glaringly obvious, and they stimulate useful retrospective comments from users.

4. CONCLUSION

The presented work is still on its way. We are actually in a feedback loop which is triggered from the experiences in the real-world applications of sonicson to go again into deeper details in research and vice versa to explore new research results in thrilling commercial applications. It will be still a long way to reach all of the mentioned aspects in a satisfying way of high-quality results. Nevertheless we achieved with the presented approach very promising effects to build real-life applications having usable convenient interfaces.

Our future work will remain two folded. State-of-the-art research results will be evaluated in real-life scenarios. The system architecture of BEAGLE is flexible enough to incorporate new agents and enhancements of the underlying knowledge representation. Especially the latter is open for different formats which seems very important to stay on the track with future developments in the scope of MPEG-7, 21, resp. the semantic web.

5. ACKNOWLEDGMENTS

Our thanks to DFKI for allowing us to work in the most creative and inspiring working environment for AI in germany. Additionally, thanks to Prof. Jameson for the hints to usablitiy design of our interfaces.

6. REFERENCES

[1] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, W. R. Swartout: *Enabling technology for knowledge sharing*, in AI Magazine, 12 (3), 1991,36-56

[2] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, & M. A. Musen. *Creating Semantic Web Contents with Protege-2000*. IEEE Intelligent Systems 16(2):60-71, 2001

[3] Pachet F., Cazaly D., *A Classification of Musical Genre*, Proceedings of Content-Based Multimedia Information Access (RIAO) Conference, Paris, France, 2000

[4] Gracenote Website (former CDDB): www.gracenote.com

[5] Allamanche E., Herre J., Hellmuth O., Froeba B., Kastner T., Cremer M., *Content-based Identification of Audio Material Using MPEG-7 Low Level Description*, Fraunhofer IIS-A, Germany

[6] FreeDB Website: www.freedb.org

[7] French J.C., Powell A.L., Schulmann E., *Automating the Construction of Authority Files in Digital Libraries: A Case Study*, Technical Report, University of Virginia, 1997

[8] CSIRO Maate Toolset: www.csiro.au

[9] Wang Y., Vilermo M., *A compressed domain beat detector using MP3 audio bitstreams*, Proc. ACM Multimedia 2001, Sep.30-Oct 5, Ottawa, Ontario, Canada, pp. 194-202, 2001

[10] Wahlster W. (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer, Berlin, 2000

[11] Cormode G., Sahinalp C., *Obliviously Approximating Sequence Distances,* Report, Case Western Reserve University, Cleveland, OH, 2000

[12] Luzeaux D., *String Distances*, Distancia92, Rennes, France, 1992

[13] Ramakrishna M.V., Zobel J., *Performance in Practice of String Hashing Functions*, Proc. DASFAA Conference, Melbourne, 1997

[14] Wagner, R. A., Fischer M. J., *The string-to-string correction problem*, J ACM 21, p.168-173, 1974

[15] Gadd T., *PHONIX: The Algorithm*, Program, 24(4), p381-402, 1990

[16] Pfeiffer S., Vincent T., *Formalisation of MPEG-1 compressed domain audio features*, Technical Report 01/196, December 2001, CSIRO Mathematical and Information Sciences, Australia

[17] Joachims T., *Text Categorization with Support Vector Machines: Learning with many relevant Features*, Proceedings of the 10th European Conference on Machine Learning, pp. 137-148, Chemnitz, Germany, April 1998

[18] Autonomy, *www.autonomy.com/autonomy_v3/ Content/Technology/Background/TheDRE*

[19] Insiders AG, *www.im-insiders.de/produkte/mindaccess.htm*

[20] SER, *www.ser.de*

[21] Junker M., *Heuristisches Lernen von Regeln für die Textkategorisierung*, Dissertation Universität Kaiserslautern, 2001

[22] Schneider G., *Mobiler, sprachgesteuerter Zugang zur Musiksuchmaschine BEAGLE,* FB Multimedia & Medieninformatik, Fachhochschule Trier