Editorial Metadata in the Cuidado Music Browser: Between Universalism and Autism

Amaury Laburthe, François Pachet, Jean-Julien Aucouturier
Sony CSL - Paris
amaury | pachet | jj @ csl.sony.fr

Abstract

We address the metadata management problem in the context of future Electronic Music Distribution (EMD) systems. We propose a classification of existing musical editorial systems in two categories: the autistic and the universalists. Universalists propose shared information, at the expense of consensuality. Autistic approaches allow individual parameterization, at the expense of reusability. We propose an architecture and a system for managing editorial metadata that lies in the middle of these two extremes viewpoints: we organize musical editorial information in such a way that users can benefit from shared metadata when they wish, while allowing them to create and manage a private version of editorial information. A mechanism allows to synchronize both view (the shared and the private view). We describe the architecture and the application in progress, in the context of the Cuidado European IST project.

1. Introduction

1.1 Vision

Picture yourself on a trip to Iceland. Before leaving, you have got hold of a Bjork mp3 from a friend. You have loaded the file on your mp3 walkman, which already contains about 5,000 of your favorite tunes. Each song on this walkman is associated to metadata: about the artist, the title, etc. Although you enjoy the new tune a lot, you don't know much about Bjork: you have only been able to classify the artist in your walkman's local genre taxonomy as "Electronica /Icelandic".
As you walk by a café in Reykjavik, you decide that you would like to see which information the local community has to offer about Bjork (she's Icelandic, isn't she ?) You turn on your walkman which immediately connects to the nearby metadata server at the café and download locally available information. Unfortunately, access to the music files is copyrighted and restricted. However, you have access to the corresponding metadata. Your walkman automatically browses the local genre taxonomy, and is able to trace down Bjork, although she's typed in with her full Icelandic name, Gudmundsdottir, and classified in a different genre: Pop/Trip Hop. The local server has a lot of available metadata about Bjork. Your walkman's metadata manager is able to fill in the fields corresponding to album, track listing, recording date and also downloads the fact that Bjork was a member of another Icelandic band, the Sugarcubes. Your friend back home will certainly be interested in knowing this! However, you decide to preserve your carefully-built genre taxonomy, and do not update your local genre metadata, nor Bjork's full name: it is only useful on the local server because it also holds some mp3s of classical music by a female pianist named Anna Gudmundsdottir.
You later notice that the local server has a metadata field which does not exist in your metadata manager: the mood of the song. You realize how useful this would be to browse your own collection. Your walkman matches the many songs that you have in common with the server, creates a new field in your metadatabase, and uploads the mood information. While you're at it, you also download information about the many other songs by Bjork that sit on the server. The next thing you do is to connect to your favorite online music shop, and buy all Bjork songs which have the same mood as the one that sits on your walkman. As your shop's server does not support browsing by mood, you decide to contribute to their metadatabase by submitting the mood metadata about your songs. In reward, this credits your point card, and allows you to download one extra song for free: why not try this "Sugarcube" thing ?

1.2. EMD and Editorial metadata

The notion of musical metadata is now well established as a key ingredient of Electronic Music Distribution systems. To manage collections of music titles, either personal or on-line, an application must have access to many information to identify, categorize, index, classify and generally organize music titles.

Because there are so many types of information that can be made explicit about music titles, musical metadata comes in many flavors. However, classifying musical

metadata based its ontological nature is a difficult task, because there is virtually no limit to what can be said about a music title. In this context, we are interested in metadata which has the following properties:

- It is *useful*, i.e. corresponds to actual features of the applications targeted.
- It is *consensual*, i.e. the information makes sense to a large part of the targeted audience, and these people would usually agree on them.

The distinction between the various forms of musical metadata is usually made based on the way this metadata is extracted, adopting a engineering viewpoint. Not only is this approach easier, but it is probably today the only one which is reasonable. Musical metadata can be divided in the following categories in this scheme:

- *Identification information*: this information allows to characterize a music title uniquely.
- *Editorial information*: this information is related to prescriptive knowledge about the music
- *Acoustic features*: this information corresponds to objective, acoustic features of the music titles. It is normally extracted automatically from the signal.
- *Cultural information*: this information captures similarity between music that emerges from socially shared sources of textual information, such as web search engines.

The IST Cuidado Music Browser project consists in designing and implementing a music browser that gathers all these kinds of metadata [1]. One exemplar feature of the Music Browser is that it implements the complete chain linking music titles seen as objective items (signals or texts) to users, considered as complex subjects. Moreover, we place ourselves in the emerging context of local and mobile ad hoc networks [2,3]. Ad-Hoc networks are wireless, self-organizing systems formed by co-operating nodes within communication range of each other that form temporary networks. In such environments, different users, with different goals, share the resources of their devices, and form an open community.

Furthermore, this paper focuses on editorial metadata, as it is designed and used in the Cuidado Music Browser. The other dimensions of the project are described in other papers [4].

2. Existing Editorial Metadata Information Systems

2.1. Autists vs universalists

Editorial metadata is today no longer a fantasy: they crops in virtually every musical application. There are, however, two radically opposed approaches in how this metadata is organized:

1) The "Autistic" approach consists in letting individual users handle their metadata in isolation, with very limited sharing. This is the approach of most peer-to-peer systems today, such as Kazaa.
2) The "Universalist" approach consists in creating a central, shared database server that all clients feed from. Examples of this approach are AllMusicGuide (AMG) [5,6], MusicBrainz [7,8] or MoodLogic[9].

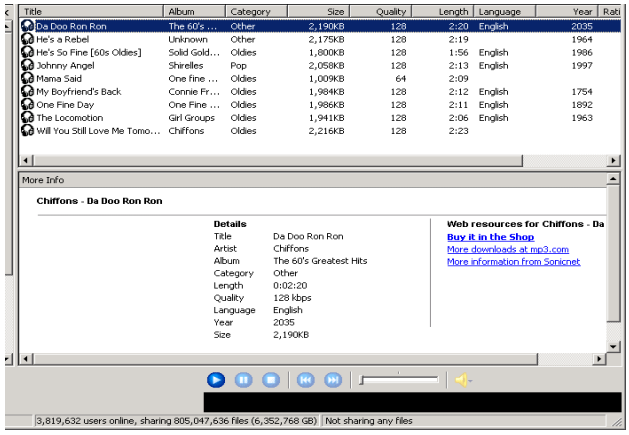


Figure 1 – Kazaa management of ID3 tags

The autistic approach consists in letting users manage themselves editorial information. A system such as Kazaa (see Figure 1) proposes different fields based on ID3 tags for describing music titles. The fields are: Title, artist, album, category (corresponding to genre), and year. Additionally you can add the language, some keywords and a short description of the track. This approach has an obvious drawback in terms of useability: user must painstakingly fill the fields for all the new titles they enter in their collection. There is almost no sharing of this metadata, other than through the actual transfer of files: When user A downloads a file from another user (B), he also gets the associated metadata.

This metadata can itself be non compatible with existing metadata. For instance, user A may have decided to spell an artists as McLaughlin, John, while B spells it John McLaughlin. A has therefore to change manually all the artist metadata of the downloaded files.

The universalist approach aims at suppressing this drawback, by imposing fixed metadata. A central server contains the metadata for a certain number of songs. When a user decides to annotate a file, a query is made to the server, together with a signature of the file. The server identifies the file from the signature, and provides the required metadata. While this approach does avoid manual annotation, it also has a price: 1) the metadata is fixed, and imposed, and the user cannot change it, 2) it works only to the extent that the signature database of the server actually includes all the music files of your collection.
The editorial information system we propose lies in the middle of these two extremes: we allow both sharing of information through a central server, and at the same time local personalization. In the next sections we will detail the nature of the metadata managed, and the client server architecture of the system.

2.2 Two kinds of Editorial Metadata

Editorial information servers such as MusicBrainz, Moodlogic or AMG provide two sorts of metadata:

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

The first category does not raise any particular problem, as this information is universal by nature. It includes for instance:

- Artist and songs name (AMG, Moodlogic & Musicbrainz),
- Albums and tracks listing (AMG & Musicbrainz),
- Group members (AMG),
- Date of recording for a given title (Moodlogic),
- Short biography for artists with Date of birth, Years of activity (AMG),
- Albums plus sometimes album credits (AMG),
- Label (AMG),
- Charts & awards (AMG with Bilboard.com).

However, these information are not particularly useful for content-based search systems such as the music browser, which aim at matching music titles with tastes: tastes, whatever they are, are rarely well expressed using administrative information on music.

The second category is both more interesting and problematic. Content description include such widely needed information as:

- Artist style (AMG),
- Artist instruments (AMG),
- Song mood (Moodlogic),
- Song review (AMG),
- Song or Artist Genre (AMG, Moodlogic),

and more generally attributes aiming at describing the intrinsic nature of the musical item at stake (artist or song). These descriptions, again, are useful to the extent that they can be used for musical queries in large catalogues. The user tests performed in the Cuidado project showed that there is virtually no limit to such information. As explained in the next section, we have added several more of such descriptors in the Music Browser. Moreover, we propose here an open approach where the user can adapt/add any descriptor to suit his needs or tastes.

In conclusion, the existing approaches cover only the two extreme cases: editorial metadata which is universal, and shared by the whole world (AMG, MusicBrainz), or metadata which is unique to their author (peer-to-peer systems), and transmitted on a file-by-file basis.

We propose here an intermediate approach, which covers the case where users want both to share content editorial metadata, and yet be able to express their own vision of the world by adapting it locally.
In the next section we describe the nature of editorial metadata managed by the Cuidado Editorial Information Manager, the choice made for music title identification and describes the architectural issues raised by the management of private and shared information.

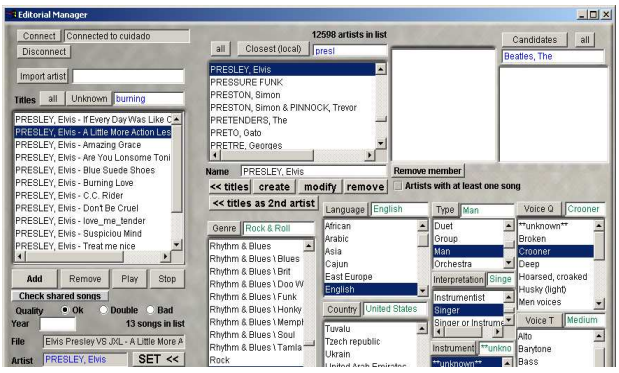
3 Editorial Metadata in the Music browser

3.1 The cuidado music browser

The cuidado music browser aims at exploiting all possible metadata that can be extracted or accumulated for music titles. The architecture we present here is focused on the management of editorial data described in section 3.2 and 3.3. Therefore we will focus on the editorial metadata manager integrated in the music browser. The kind of Editorial metadata we are interested in the Cuidado Browser is metadata that can be used readily for searching music. More precisely, our editorial metadata appears directly under the form of search fields that can be used in the browser. Figure 2 shows the query panel of the Music Browser, in which several editorial metadata information is displayed and can be used to issue musical queries.

3.2 Editorial information about titles

As seen on figure 3, editorial information are managed with a specific tool proposing choice lists for each property. Concerning music titles, our tool enables basic editions as title name or keywords, as well as less obvious features such as title genre, primary and secondary artist.
The notion of primary and secondary artist has been introduced to represent the various degrees of association between artists and music titles in a generic way: what is important for a musical query system is not necessarily to make the distinction between all possible roles of artists (composer, performer, conductor, remixer, etc.), but to propose a simple indexing scheme. In all cases, the Cuidado editorial information manager proposes an unified view of artists link to songs as "primary" and "secondary". These notions of primary and secondary have different significations according to the context: In Popular music, performers are usually put forward for identifying music (e.g. With a Little Help from my friend by "Joe Cocker"), and composers come last (e.g. with a little prayer sung by Aretha Franklin (primary artist) is in fact composed by Burt Bacharach (secondary artist). In classical music, the distinction is inverse: for example the Opera Rinaldo is primarily identified with Haendel (composer). A user may want to access a particular recording of this Opera by conductor René Jacobs (secondary artist). In another context, some remixes of songs can be identified primarily by the remixer: The recent remix of the song "A little less conversation" is primarily identified JunkieXL (a famous remixer). In second approximation, this song is an Elvis Presley song (i.e. usually performed by Elvis Presley).



3.3 Editorial information about artists

On top of the artists metadata already described in section 2.2, the Cuidado Editorial Information manager adds some content features deemed useful for browsing, and not present in any existing editorial server:

- Type : Michael Jackson is a *singer* while the Beatles is a *band*, Elvis Presley is a *singer* and a *musician*, while JunkieXL is a *DJ* and a *remixer*,
- Interpretation: The Beatles have mainly recorded *music with vocals*, while John Coltrane has mainly played *instrumental music*,
- Voice quality: Frank Sinatra is a *crooner*, while Janis Joplin has a *broken voice*,
- Voice range: Barry White has a *bass* range, while Britney Spear has a *soprano* range,
- Language: The Beatles sing in *English*,
- Keywords: any other relevant information, such as "1975 *live version*", "*remix*", "*birthday song*", …

Moreover the Cuidado Editorial Manager proposes some semantic information about artists. For instance, many artists belong to groups: Paul McCartney belongs to The Beatles, Phil Collins to Genesis, etc. This information is not only useful for administration purposes, but can also readily be used for browsing. We introduced the "memberOf" predicate in the Cuidado editorial database. Figure 4 shows an example of the use of this information.

Figure 4 – the "member_of" predicate.

4. Music Identification

One common element of every EMD system is a front-end able to link musical files, either available on the user's devices or online, to the metadata describing the corresponding music object – or *music title*. Although this identification stage is independent of the management strategy of the whole system, and may be found either in the peer-to-peer, universalists or ad-hoc approach, it is nevertheless an essential component of the metadata management chain. In this section, we describe the choice made in the Cuidado Music Browser.

4.1. Content-Based identification

Identification can be done in a blind way simply by analysing the music signal. Over the past few years, there has been a great deal of academic and industrial efforts concerning this technique, usually referred to as *Audio/Music Fingerprinting* or *Hashing*. The general idea is to extract a very compact representation of the music signal, its *signature* or *fingerprint*, and to compare it to a database of already extracted and identified signatures. The signatures should be very robust to noise, so that many distorted/compressed/broadcast/variously encoded instances of the same music title can be matched to one unique entry in the database. Signatures should also be compact, so that matching one test signature against a huge database (usually several tens of thousands signatures of songs) can be done in feasible time.
Different indexing schemes and search algorithms are then used to match the extracted signature against the database.

The reported performances of the various identification algorithms all are very good, usually in the top-1% using realistic levels of noise and distortion. This makes these technologies well-suited for many commercial applications. The business model used by Moodlogics [9], ID3Man [10] (with its fingerprinting technology Auditude [11]), MusicBrainz [7,8] (with Relatable [12]) , Tuneprint [13], GraceNote [14] (integrated into Apple's iTune and mp3 walkman iPod) allows users to link their personal music files to metadata that has been gathered on a server by the provider. One common extension of this is to automatically fix the ID3 tags of the user's mp3, or even to rename the files themselves with their correct title and artist name as identified from the database.
Other commercial, much advertised applications of fingerprinting technologies are Broadcast Monitoring (Yacast [15]), Filtering technology for file sharing (e.g. preventing copyrighted files to be exchanged in Napster) or "Name that tune" applications on mobile phones (Shazam [16]) or on digital radio on PC (Clango [17]).

The audio fingerprinting approach is well suited to the universalist approach, in which it is considered implicitly that the collection of titles is finite and shared by all. In our context, we target communities of users who do not necessarily access files that are sufficiently well known to be included in the signature databases. Furthermore, communities may wish to specialize in specific musical areas, including share metadata on music titles which are not produced by majors.

4.2. Using external information

Another way to identify music files and link them to metadata consists in using external information on the titles when available.
For instance, the Sony's Emarker system (discontinued in September 2001, see [18]), used to exploit the geographical and temporal location of a radio listener requesting a song, and then query a large database containing all radio stations programs by time and location. The approach is of course much lighter than the signal based approach since no signal processing is required, and can scale-up to recognize virtually any number of titles. It works of course only for titles played on official radio stations.

External information can be as simple as file names, with the difficulty that names are even less standardized: an artist such as "The Beatles" may be catalogued as "The Beatles", "Beatles, The", or any other combination. In [19], we have proposed a heuristic-based parsing system to exploit the information possibly contained in the file name itself. We have studied large corpora of files, whose names are decided by humans without particular constraints other than readability, and have draw various hypotheses concerning the natural syntaxes that emerge from these corpora. A central hypothesis is the local syntactic consistency, which claims that file name syntaxes, whatever they are, are locally consistent within clusters of related music files. These heuristics allow to parse successfully file names without knowing their syntax a priori, using statistical measures on clusters of files, rather than on parsing files on a strict individual basis.

For instance, it is impossible for an automatic system to parse a filename like:

```
D:\mp3\CSL2-9\Various – RockFM
\Original Rock – 5 - Crack The World
Ltd - Fine Young Cannibals - She Drives
Me Crazy.mp3
```

To start with, which section is the artist name ? "Rock Fm", "Original Rock", "Fine Young Canibals", or "She drives me crazy" ?

However, we can observe that in the same directory, there are many filenames having the same syntax

`a-0-b-c-d,`

where a,b,c,d are strings and 0 is a number, and then look at the statistics on the different sections :

- a and b are always the same ("Original Rock" and "Crack the World Ltd")
- 0 is incrementing
- there are several different d's for each c ("Fine Young Canibals – She drives me crazy", "Fine Young Canibals – Good Thing", …)
- there are usually more words in d than in c
- etc.

From all these statistics and with a few appropriate heuristics, the algorithm is able to infer that c is the artist field and d is the song title field. Experiments in [19] have shown that the parsing error with this algorithm is below 5%, which compares with the recognition rates achieved by fingerprinting techniques.

It is this second approach that we have chosen to use in the Music Browser. The audio fingerprinting approach is well suited to universalist approach, in which it is considered implicitly that the collection of titles is finite and shared by all. In our context, we target communities of users who do not necessarily want to access files that are sufficiently well known to be included in the signature databases. Furthermore, communities may wish to specialize in specific musical areas, including shared metadata on music titles which are not produced by majors. Finally, we deemed that maintaining very big databases of fingerprints was not suitable on small devices aimed at local or ad-hoc networks.

5. Architecture of the system

This section describes the client server architecture underlying the management of editorial information in the Cuidado music browser.

5.1 Architecture of the Cuidado editorial information manager

As shown in figure 5, the CUIDADO metadatabase is a MySQL database hosted on a SQL server. The server acts both as a server for Php scripts and servlets. The MusicBrowser is implemented in Java and communicates with the MySQL database using JDBC drivers. The editorial metadata server runs a Php server accessible over the Internet. Specific Php scripts allow client applications to fetch and submit editorial metadata to this server.
However, Php scripts are not efficient enough to handle a variety of operations. In particular operations requiring large amounts of information to be loaded in memory. To address this issue, the Cuidado server includes a servlet server. For artist and title identification, this servlet load precompiled information in memory (typically the list of artist and title names) to speed up approximate string matching algorithms based on Levenshtein distance.

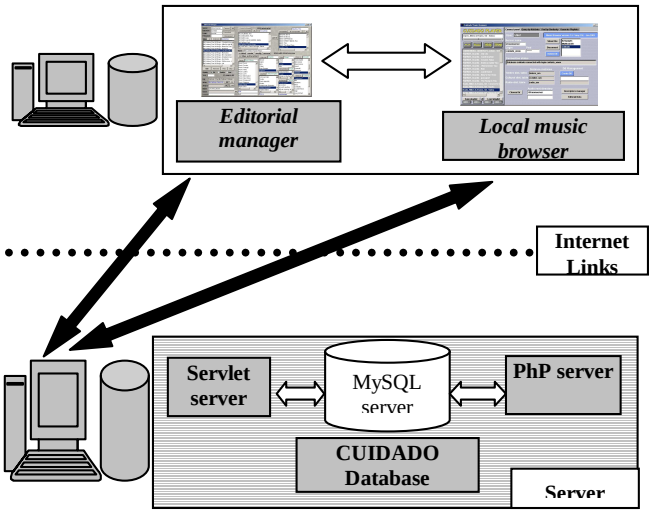


Figure 5 - Editorial data management

Note that such an architecture uses only free and standard middleware components. The music browser as well as our architecture runs on windows, Mac and Linux machines in a transparent way. As shown on figure 6, a community can run a server on a local or ad-hoc network with possibly different metadata attributes than the central server.

5.2 Local vs Shared metadata

With the apparition of ad-hoc networks, single or multiple users can share their data easily and in a transparent way. This situation raises a key issue: the management and synchronization of the data. How can users keep their database up-to-date while benefiting from new entries without degrading their customized databases.

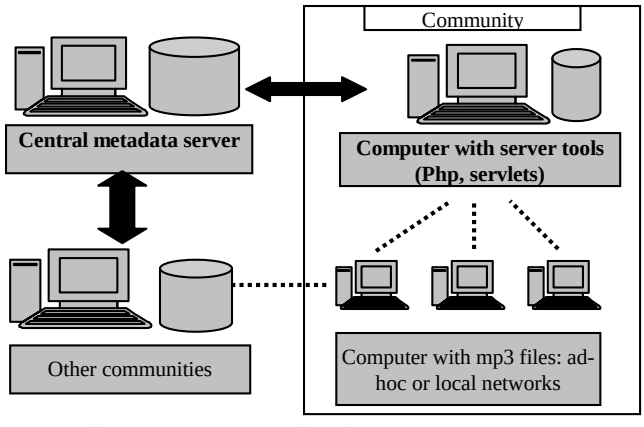


Figure 6 - Interaction between systems

We propose an architecture that addresses this problem, while still being based on a central server architecture (Figure 6). In our architecture community users all work on a community server, itself synchronized with the central server. The music browser is installed on each users' computer and is used as a front-end to create/modify metadata. This architecture is based on two main operations: *update* and *infer*, which are described in the next section.

5.3 Updating metadata (client side)

5.3.1 Adding new songs or artists. Users can add new songs and/or artists to their local database, using the data management tool (figure 3). User choose song files (e.g. mp3, wav, etc.) or enter artist name manually and the corresponding file/artist names are automatically analyzed. As described in section 4 we use a parsing mechanism to automatically recognize artists and song names. Using a tuned Levenshtein distance, the client metadata manager looks for artists and songs in the local database as well as in the central server database. Three cases are possible:

- if the song and/or artist exists on the central server the user is proposed a list of closest matching artists and/or songs and can link his new songs with the chosen one while all data are imported. This process ensures that every database shares the same artist and song indexes to avoid compatibility problems. If Michael Jackson is referenced as artist #98 and a memberOf Jackson5 referenced as artist #10, then every local database must use these same indexes. *See reference 1 on figure 7.*
- If the song and/or artist already exists in the user's local database, then the new entry can be removed (to avoid double), marked as double, or as a new song and/or artist (e.g. for a live version or homonym artists). *See reference 2 on figure 7.*
- If the song and/or artist does not exists at all, users create all metadata using the management tool. These data are stored on the local server and broadcast to the central server for further processing. *See reference 3 on figure 7.*

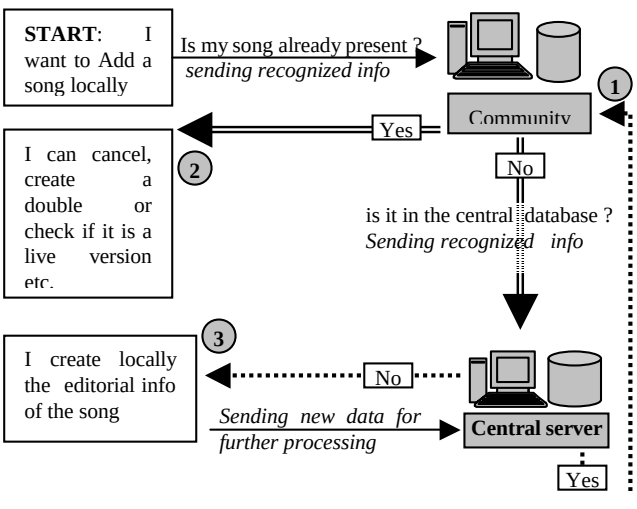


Figure 7 – Adding and importing new data

Music is constantly evolving and no system could reasonably forecast everything [5]. Furthermore as communities can run their own metadata server they will most probably want to tune it to create new fields or simply to add a new musical genre not yet present. The Cuidado editorial manager includes such a feature, i.e. the ability to update the database structure itself: users can update their database structure to evolve it. As for songs and artists, such modifications are broadcast to the central server for further processing.

5.3.2 Updating the local metadatabase. It can happen that a song or an artist in the central server is created or modified. The metadata manager has the ability to synchronize local metadata with the shared metadata of the server. When wanted, users choose to update a part or all the local metadata. The same mechanism is available for the database structure.

5.4 The infer process

When new data are submitted to the central server, they need to be integrated. We call this the infer process. It is envisaged here in a collaborative filtering way [20]. Data are stored and regularly analyzed by the central server. Emergence of consensus enables to consolidate new entries. This process is performed automatically to avoid manual moderation which is a time-consuming process. Once a week metadata are updated on the central server. When a community user performs a synchronization all local data are updated. Each community server using the central one, at least for the basic indexes, the compatibility is ensured. However they always have the opportunity to refuse updates, new entries considered as non relevant for the community, etc. As in Musicbrainz the central server will benefit from users entries (although the MusicBrowser already performs pretty well as a stand-alone software to manage large collection of music files).

Gathering data being a key issue for most metadata systems we believe that community vision can represent an interesting new approach. Shared among specific music genre specialists (people involved in "East Coast Rap" or in "Intelligent Techno" using an ad-hoc network) a database can quickly become highly specialized with a

limited number of users. Members will probably be more keen on adding and consolidating entries if they see an immediate benefit for their community. This community then share their data with the central database server without degrading their data and without necessarily opening their database to everybody.

Conclusion and future works

In the context of ad-hoc and local network based communities, users want both to share metadata, and simultaneously manage metadata of their own. We presented an architecture for managing musical editorial metadata which allows client applications to exploit shared metadata when available as well as creating and managing local, private information. This architecture is based on two basic principle: an update mechanism, which warns the central database of any local modifications, and an infer mechanism, which computes emerging, consensual values from user inputs. The resulting architecture provides greater flexibility in editorial metadata management for electronic music distribution systems.
This system is a first step in the direction of hybrid metadata systems, in the sense that it lies between the two extremes of universalist and autist approaches. Current work focuses on extending this paradigm to include other forms of metadata, in particular acoustic metadata computed from the audio signal, as well as musical similarity relations computed from data mining techniques. Finally, user experiments are in progress to assess the robustness of our approach in the context of the IST Cuidado project.

References

[1] F. Pachet, A. La burthe, J.J. Aucouturier, A. Zils "The Sony Popular Music Browser" JASIST 2003

[2] Mesh networks web site http://www.meshnetworks.com

[3] Many-to-Many Invocation M2MI : A New Object Oriented Paradigm to build Software Infrastructures for Serverless Ad Hoc Collaborative Systems http://www.cs.rit.edu/~hpb/Talks/M2mic_OS/all.pdf

[4] H. Vinet, P. Herrera, F. Pachet "The CUIDADO Project : New Applications based on audio and music content description", Proceedings of ICMC, pg 450-454, 2002. ICMA

[5] D. Datta, "Managing Metadata", ISMIR 2002, Paris, France, October 2002

[6] AMG, Allmusicguide web site http://www.allmusic.com

[7] MusicBrainz website http://www.musicbrainz.org/

[8] A. Swartz "MusicBrainz: A semantic web service"

[9] Moodlogics website http://www.moodlogic.com

[10] ID3Man website http://www.id3man.com/

[11] Auditude website http://www.auditude.com

[12] Relatable website http://www.relatable.com

[13] Tuneprint website http://www.tuneprint.com

[14] Gracenote website http://www.cantametrix.com

[15] Yacast website http://www.yacast.com

[16] Shazam website http://www.shazamentertainment.com

[17] Clango website http://www.clango.com

[18] Sony's Emarker story http://www.bricklin.com/emarker.htm

[19] Pachet, F., Laigre, D. "A Naturalistic Approach to Music File Name Analysis" in proc. 2nd International Symposium on Music Information Retrieval, Bloomington Indiana, USA, October 2001

[20] Sarwar, Karypis, Konstan, Riedi "Item-based collaborative filtering recommendation algorithms" ACM 2001