A Naturalist Approach to Music File Name Analysis

François Pachet
Sony CSL-Paris
6, rue Amyot
75005 PARIS
France
pachet@csl.sony.fr

Damien Laigre
Sony CSL-Paris
6, rue Amyot
75005 PARIS
France
dlaigre@csl.sony.fr

Abstract:

Music title identification is a key ingredient of content-based electronic music distribution. Because of the lack of standards in music identification – or the lack of enforcement of existing standards – there is a huge amount of unidentified music files in the world. We propose here an identification mechanism that exploits the information possibly contained in the file name itself. We study large corpora of files whose names are decided by humans without particular constraints other than readability, and 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. Based on these validated hypothesis we propose a heuristics-based parsing system and illustrate it in the context of an Electronic Music Distribution project.

1 Introduction

The recent progress of digital audio technologies and the availability of easy and cheap Internet access have led to the proliferation of music files on the planet.

Efficient digital audio compression format such as mp3 have made possible the distribution of music on a large scale, using all sorts of broadcasting techniques and supports, such as peer-to-peer communication systems. This proliferation of music data around the globe is not incidental, and may be seen as a sign of the huge pressure for Electronic Music Distribution (EMD) from the community of music listeners.

EMD, however, is more than just representing music as audio files. Confronted to large databases, users can only access what they know, and content-based management techniques are acknowledged to be a necessary ingredient to fulfil the target of true, personalized music distribution.

Content-based music access requires, between other things, the ability of extracting features from the signal, of gathering descriptions of various source of textual information, of modelling user profiles and matching these profiles to music descriptors, etc. (see Pachet, 2001a for a survey). Among these requirements, one key issue is music identification: how to identify in a non-ambiguous way music files. This identification is crucial to allow the management of metadata, copyrights, profiles, recommendation systems, etc. Without a solid identification mechanism, EMD may well turn into a gigantic and serendipitous adventure for users, content providers and distributors.

Various standardization efforts have been conducted to define universal codes for music titles. The most famous is probably the ISRC (International Standard Recording Code), developed by ISO (ISO 3901) to identify sound and audio-visual recordings. ISRC is a unique identifier of each recording that makes up an album. Unfortunately it is not followed by all music production companies, and hardly used in unofficial music sources such as peer-to-peer communication systems.

Another problem is that, even when a code could be used, it is not: for instance, digital music encoded in the audio CD format usually does not contain information on the music identification. Strangely enough, it is not possible to get the track listing information from a CD. External databases of track listings for commercial CDs have been developed, such as CDDB. CDDB works by associating track-listing information to audio signatures of CDs. To allow scaling up, CDDB is a collective effort: the database is made up by the users themselves. While this collaborative aspect does allow scaling up (there are more than 4 millions CDs registered on CDDB), there is an obvious drawback to this enterprise: the track listing information is not guaranteed, which leads to many errors, duplications and to the difficulty of identifying correctly music titles.

There are other sectors of the music production chain that are concerned with music title identification, such as radios (which display their track listing on Internet for instance) or copyright associations (which have to keep track of broadcasted titles to compute the payment of royalties). In each case, ad hoc and proprietary schemes have been devised, but there is no convergence of music identification methods.

There are several approaches to the identification of audio music sources. The most straightforward one consists in analysing the signal, typically a portion of the whole music title, to extract an audio signature. This signature is then matched against a database of pre-recorded music signals. This task is, for instance, addressed by technologies such as Broadcast Data Systems (US) or MediaControl (Germany), and is used by copyrights management companies to infer radio play lists. The techniques used to perform the identification range from usual pattern matching to more elaborate statistical methods based on characterization of the evolution of spectral behaviours. In all cases, the identification requires a database of all music files created beforehand. Such a global database is far from realistic in the near future sot the approach can work only within limited contexts.

Another approach consists in exploiting external information about the music source. For instance, the Emarker system (Emarker, 2001), exploits the geographical and temporal location of a radio listener requesting a song, and then queries 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.

In this paper we describe another approach, more suited to personal music file management systems, for which no radio track listing is available for identification, and which does not require the management of a global, universal database of music titles. This approach is based on the analysis of actual music file names.

More precisely, we consider the context of popular music titles, and therefore seek to identify two main information for a music source: the artist (or performer) identification, and the actual name of the music title. We consider music file names coming from natural sources, such as personal hard disk drives (usually filled with audio files coming from peer-to-peer communication systems), track listing databases (such as CDDB), or radio track listings. In all these cases, the file names are input by users who do not follow any constraint, other than human readability.

We consider here information contained in music file names, and not identification from the signal, or from other external sources of information (such as ID tags in mp3 files, see Hacker, 2000). These other methods are orthogonal to the method proposed here. In an ideal case, music identification could exploit all these methods collaboratively.

We will first introduce the context of our study, and the corpora analysed (Section 2.1). We then propose several assumptions for guiding the analysis process, the main assumption being a local consistency assumption (Section 2.2). We perform a statistical analysis of these corpora to validate the assumptions and draw corresponding heuristics. Finally, we describe FNI, a system that implements our heuristics, and illustrate how it performs in the context of a real world Electronic Music Distribution system developed at Sony CSL, within the European CUIDADO IST-funded project.

2 Popular Music file names

Music file names may contain various types of information about a music title. In our context we focus on popular music, for which two information are of interest: the artist or interpreter identifier, and the actual title name. In some cases, file names can also contain other information such as the album or track number. In the case of Classical music, the notion of artist is more complex, and identification may contain both composer and performer identifier. Additionally, various identifiers may also be present, such as the version (instrumental, remix, etc). Several statistical approaches have been proposed to parse text automatically into coherent segments, corresponding for instance to different topics in news transcripts (see e.g Beeferman et al., 1999). In our case, the textual data considered is much shorter and the domain (music works) is narrower, so we show that it is possible to derive heuristics to implement an efficient parsing system without a learning component, at least as a first approximation.

2.1 Corpora studied

Music files names are typically found in the following locations: 1) personal storage systems such as hard disks, 2) radio program track listings and 3) repositories of musical metadata. For the purpose of our study, we identified three such databases: a subset of 22, 302 album track listings from the CDDB database containing track listings for about 4 millions CD albums, 2) a 1-year listing of a radio station broadcasting music in a large variety of styles (Fip/RadioFrance) and 3) a set of file listings (about 3000 files) of personal hard-disks of intensive users of peer-to-peer music communication systems.

These three cases share a common characteristic: the file names they contain have been specified by individuals on whom no particular syntactic constraint was enforced, other than human readability, i.e. the fact that these names should be understood easily by other individuals of the same community. The individuals name their files as they wish, and these personal conventions are simply spread through the community without modification. In CDDB, the principle of the database is collaboration: albums track-listings are given by the users themselves. Although the editors for entering track-listing information may in some case force some structure (e.g. differentiate title and artists), there is no unique syntax valid for all track listings, as illustrated below. In the case of radio stations broadcasting their programs, there may more be cohesion since these programs are entered by a smaller number of individuals, but, similarly, the syntax will not be constant, and will differ from a radio station to another one. However, the case of radios is simplified by the fact that the syntax of file names is usually constant for a given radio.

To illustrate our study, we give below some typical examples of file names coming from the sources at hand.

```
The Rolling Stones - Angie
The Beatles - Oh! Darling
Eagles - Hotel California
Simon & Garfunkel - The Sound Of Silence
Kansas - Dust In The Wind
America - The Last Unicorn
Creedence Clearwater Revival - I Put A Spell On
You
The Beatles - Let It Be
The Tremeloes - Silence Is Golden
Hollies - He Ain't Heavy He's My Brother
Zz Top - Blue Jeans Blues
Simon & Garfunken - El Condor Pasa (It I Could)
Bee Gees - Massachusetts
Omega - The Girl With The Pearl's Hair -
Featuring Gabor Presser, Ann
```

Figure 1. File names found on the CDDB database, for an album entitled *"Golden Rock Ballads V.1"*

```
d:\mp3\CSL2-1\Various - Animals - The house of
the rising sun.mp3
d:\mp3\CSL2-1\Various - The Mindbenders - A
groovy kind of love.mp3
d:\mp3\CSL2-1\Various - Hollies - The Air That I
Breathe.mp3
d:\mp3\CSL2-1\Various - The Beatles - Ain't she
sweet.mp3
d:\mp3\CSL2-1\Various - Bee Gees -
Massachusetts.mp3
d:\mp3\CSL2-1\Various - The Moody Blues - Nights
in white satin.mp3
d:\mp3\CSL2-1\Simon and Garfunkel – El Condor
Pasa (If I Could).mp3
d:\mp3\CSL2-1\Simon and Garfunkel – The Sound of
Silence.mp3
d:\mp3\CSL2-1\Bee Gees - Saturday Night
Fever.mp3
d:\mp3\CSL2-1\Beastie Boys - Song for Junior.mp3
d:\mp3\CSL2-1\Beach Boys - Good Vibrations.mp3
d:\mp3\CSL2-1\01 - The Beatles - Doctor
Robert.mp3
d:\mp3\CSL2-1\05 - The Beatles - Sgt Pepper's
Lonely Hearts.mp3
d:\mp3\CSL2-9\Various - Rock F.M\Original Rock
N°5 - Crack The World Ltd - Fine Young Cannibals
- She Drives Me Crazy.mp3
d:\mp3\CSL2-9\Various - Rock F.M\Original Rock
N°5 - Crack The World Ltd - The Beach Boys - I
Get Around.mp3
d:\mp3\Jazz\STAN_GETZ\MENINA_MOCA.mp3
d:\mp3\Jazz\STAN_GETZ\SAMBA_DE_UMA_NOTA_SO.mp3
```

Figure 2. File names found on a personal hard disk.

```
17:54 OH DARLING, THE BEATLES
       ABBEY ROAD (1969 EMI)
17:57 I BELONG TO YOU, LENNY KRAVITZ
       5 (1998 VIRGIN)
18:01 FATIGUE D ETRE FATIGUE, LES RITA MITSOUKO
       COOL FRENESIE (2000 DELABEL)
18:09 IT AIN T NECESSARILY SO, MILES DAVIS
       BESS (1958 CBS)
18:14 ENTRE VOUS NOUVIAUX MARIES, ALLA FRANCESCA
       BEAUTE PARFAITE / ALLA FRANCESCA (1997
OPUS 111)
18:16 FOR EMILY WHENEVER I MAY FIND HER, SIMON
AND GARFUNKEL
       COLLECTED WORKS (1966 CBS)
```

Figure 3. A typical radio program on Fip/Radio France.

```
9:28 Bach: Concerto #4 in A, BWV 1055 (Glenn
Gould, piano, Columbia SO/Golschmann) CBS 38524
9:50 Bach/Manze: Toccata & fugue in d, BWV 565
(Andrew Manze, solo violin) Harmonia Mundi
907250.51
10:04 Jaromír Weinberger: Polka & fugue from
Schwanda the Bagpiper (Philadelphia O/Ormandy)
Sony 63053
10:21 Shostakovich: Piano concerto #2 in F,
Op.101 (Mikhail Rudy, St. Petersburg PO/Jansons)
EMI Classics 56591
10:49 Dvorák: Bagatelles, Op.47 (Takács Qrt.)
London 430 077
11:14 Falla: El sombrero de tres picos (Three-
Cornered Hat), part 1 (Jennifer Larmore, Chicago
SO/Barenboim) Teldec 0630-17145
```

Figure 4. Another typical radio program on WFCR/Western New England.

2.2 Clusters

An important remark to be made is that the music files considered are usually organized in different levels. In CDDB, there is only one level which is the album, itself containing tracks. On personal hard disks, there may be any number of levels, represented by the directory structure of file systems. For the sake of generality, we consider that the database of file names is structured by clusters – possibly – recursively. Clusters may contain either other clusters of file names.

As we can see, there is no universally valid syntax, either at the lexeme level (morphology of informative elements) or the music file level (actual syntax). However, these file names are not totally random, and some regularities can be identified, in particular at the cluster level. In the next section, we examine more closely the regularities found in these various sources, from which we will draw a set of heuristics for an automatic file name recogniser.

2.3 An Empirical Analysis

A manual analysis of a subset of our databases was performed, to identify the most salient characteristics of file names. This manual analysis of some examples yields a number of regularities:

1) Regularities at the file name level. There is a small number of delimiters that are used for separating artist and title information. Based on these delimiters, there are some syntaxes with a higher degree of probability than others. For instance: "artist – title " such as "The Beatles - Oh! Darling", "title – artist" such as "Oh Darling, The Beatles", or "constant term – artist – title" such as "Various - The Beatles - Ain't she sweet", etc.
2) Regularities at the word level. Artist names are usually found under a restricted number of syntactic forms, such as: "Paul McCartney", "McCartney, Paul", "Mc Cartney", or "The Beatles", "Beatles, the", "Beatles".
3) Most importantly, regularities at the *cluster level*. It appears that syntaxes, as cumbersome as they may sometimes be, are not distributed uniformly: within a

cluster, it is often the case than all titles follow the same syntax, or, at least, a small number of syntaxes. This remark is at the core of our proposal, as we will see below.

Based on these remarks, we propose the following four hypotheses relative to music file name analysis:

Delimiter Hypothesis:
This hypothesis states that the artist and title name information are indeed separated by delimiters, which are special characters within a given, small set of characters. As a special case, we consider that a file name using no separators is a title name without reference to its artist.

Constant Term Hypothesis:
Several syntaxes may contain constant terms, which are not directly relevant. A constant term can be for instance the album name, a date, or key words such as "Various Artists" (see Figure 2). The notion of constant term here is augmented by integrating possibly varying numerals, to handle cases such as track numbers ("Track 1," Track 2", etc. see Figure 2).

Word Morphology Hypothesis:
Artist names and title names have statistically different morphologies. For instance, the number of words for artist names is less important than the number of words used for title names. Additionally, artist names often make use of a limited number of specific heuristics related to first name (McCartney, Paul" is the same than "Paul McCartney" or "McCartney, P."). These heuristics may be used to determine whether a piece of information denotes an artist or a title name.

Local Syntactic Consistency Hypothesis
This hypothesis asserts that syntaxes of file names are consistent within a given cluster (what we call a syntax will be defined more precisely below). In reality, the hypothesis is weakened by the fact that this consistency may not actually occur entirely within a cluster. For instance, Figure 2 shows a directory listing containing four main syntaxes (for a total 13 titles, which is indeed an extreme case). We weaken this hypothesis by considering sub-clusters sharing the same syntax, and showing that only a small number of sub-clusters is needed – in general – to perform the analysis correctly.

In the next Section, we show the results of an automatic analysis performed on our databases to assess the validity or our hypothesis.

3 Statistical Analysis of File Name Corpora

3.1 Delimiter Hypothesis

We call here a delimiter a character used to separate different type of information in a given segment. The hypothesis states that there are indeed delimiters: these special characters are - most often - used as separators, rather than significant characters for artists or title names. The most encountered delimiters in the corpora are the following: '-', '/', '(', ')', '[', ']', '{', '}', ';', ':'.
To validate the Delimiter Hypothesis, we have to show that the file names use delimiters to separate artist and title information. To do this systematically would require a thorough check of over 300.000 titles, which is too hard a task to be done manually. Instead, we show here that delimiters are used in a consistent manner within each cluster. Although this check does not guarantee that delimiters are indeed used to separate, e.g. artist and title information, it does a give strong indication that there is a consistent use of these characters as syntactical elements rather than significant characters.
More precisely we call "common delimiter" a character delimiter found in *all* the segments of a given cluster. This delimiter indicates in most of the case a separation between different information types. As the following table shows, many (64.4%) though not all clusters have one common delimiter. Some clusters have no delimiters (7.2%), which corresponds to cases where the file name only contains the title information (the artist name is then most often contained in the album name for CDDB, or in the super directory for personal files). In the remaining cases, several delimiters are found in given clusters. We then look for the minimum number of delimiters that "cover" the whole cluster. What the table shows is that there is, in most of the cases, a small number of such covering delimiters, which is once again a strong indication that these delimiters are used for syntactical purposes.

	Nb clusters:	Percentage:
no delimiter:	1615	7.2 %
1 common delimiter :	14354	64.4 %
2 delimiters cover the cluster :	4763	21.3 %
3 delimiters cover the cluster :	1338	6.0 %
4 delimiters cover the cluster :	215	1.0 %
5 delimiters cover the cluster :	17	0.1 %
Total :	22302	100.0 %

Figure 5 Analysis of delimiters in our CDDB play lists.

3.2 Word Morphology Hypothesis

The word morphology hypothesis asserts that artist names are shorter on average than title names. Although this hypothesis is certainly not always true (e.g. the group named "Everything but the girl" has recorded a song named "Angel"), it is true in average, and in particular within clusters.
An analysis of about 17,000 titles from CDDB yields an average of 1.6 words per artist names against an average of 3.2 words for title names, i.e. a ratio of 2 times more words in artist names. Similarly, an analysis of 19,648 titles from the FIP radio program yields 2.1 words for

artist names against 3.1 words for the title names, i.e. a ratio of 1.5.
This shows clearly that titles names are, on average, longer than artist names. As we will show below, this heuristic may be used when no other clue allows to infer whether a string is an artist or title name.

3.3 Constant Term Hypothesis

The constant term hypothesis asserts that clusters may happen to contain constant terms in *all* their segments. These constant terms can refer for instance to the artist name, but also to information which is useless in our context.
The analysis of our CDDB database yields 800 constant terms, of which about 20% are not artist names. As an indication, here are the 10 most frequent useless constant terms retrieved from this list:

```
Sampler
Various Artists
Various
Passion
Unknown
Fabulous
Success
Dreams
Memories
Mixery
```

Figure 6 Most Frequent constant terms in CDDB play lists

These constant terms are used in our system to differentiate between useless information that can be discarded from useful information such as artist names.

3.4 Local Syntactic Consistency Hypothesis

This hypothesis is the most important in our study, since it will allow us to determine the syntax according to the analysis of a group of titles, rather than individual titles only. To validate this hypothesis, we need to estimate the average number of different syntaxes a cluster contains.
To do so we introduce the notion of syntax as follows. For a given file name string, we replace all the token strings encountered by an alphabetic letter incremented automatically (a, then b, then c, etc.) and we replace all numbers by a digit (0, 1, 2; etc.). We let the delimiters unchanged. The resulting string may be seen as a canonical representation of the syntax of the file name.

Here are some examples of file names and their associated syntax as extracted by our analysis:

File name	Syntax
Various - Bee Gees - Massachusetts.mp3	a-b-c
Simon and Garfunkel – El Condor Pasa (If I Could).mp3	a-b(c)
The Beatles - 05 - Sgt Pepper's Lonely Hearts Cl.mp3	a-0-b
Original Rock N°5 - Crack The World Ltd - The Beach Boys (I Get Around).mp3	a-b-c(d)
All you need is love.mp3	a

Figure 7 Canonical syntaxes for various music file names.

As an illustration of the process, here are the most frequent syntaxes retrieved (in number of lines):

a	69277	a-b	66637
a/b	64584	a(b)c	30569
a-b(c)d	15351	a/b(c)d	19191
a:b	5561	a(b)-c	4198
a-b-c	3050	a/b-c	2508
a(b)/c	1959	a,b	1918
a-b/c	1708	a(b-c)d	1319
a[b]c	1317	a--b	1128
a/b,c	954	a,b/c	930
a:b(c)d	906	a-b,c	727
a-b[c]d	703	a:b-c	673
a,b-c	589	a/b[c]d	556
(a)b	524	a/b(c-d)e	517
a-b-c(d)e	506	a(b)(c)d	500

Figure 8 Main syntaxes found in our CDDB play lists

Once syntaxes are extracted, we compute, for each cluster, the number of different syntaxes it contains. This computation simply consists in comparing syntaxes using string comparison operators. The following table shows the result of this computation.

	Nb clusters:	**Percentage :**
1 common syntax :	4871	21.8 %
2 syntaxes in the cluster :	7702	34.6 %
3 syntaxes in the cluster :	5160	23.1 %
4 syntaxes in the cluster :	2691	12.1 %
5 syntaxes in the cluster :	1162	5.2 %
6 syntaxes in the cluster :	409	1.8 %
7 syntaxes in the cluster :	180	0.8 %
8 syntaxes in the cluster :	51	0.2 %
9 syntaxes in the cluster :	27	0.1 %
Over 9 syntaxes:	49	0.2 %
Total:	22302	100.0 %

Figure 9 Analysis of syntaxes in our CDDB play lists

These results clearly show that there is indeed a syntactic consistency in most of the clusters encountered. This consistency, in turn, will be used to parse file names according to the most prominent syntax within clusters, as shown in the next section.

4 The FileNameInterpreter (FNI) System

The hypotheses we made and validated have been exploited to design and implement a file name interpreter, in the context of an EMD application. This application is

part of *Cuidado*, a large European project for content-based music access (see Pachet, 2001b). In this section, we describe the overall design of this system, and show its performance on real world examples.

4.1 Overview

The input of our system is a file containing a structured list of file names. The output is a file containing the analysed artist and title name information. This analysis is performed by applying heuristics as described below. To allow flexibility, the user always has the possibility to correct manually the analysis proposed, and this correction is then substituted to the analysis in the output file, and retrieved in later analysis to avoid repeating corrections.

4.2 Initialization

A pre-processing phase is applied systematically to the input list of music file names. This pre-processing consists in:

1) Grouping together file names having the same syntax into sub-clusters,
2) Chunking related file names into segments according to delimiters.

For instance, if we consider the input file as given in Figure 2, considering only the first cluster we obtain the following:

1) The syntaxes found in this corpus are: `“a-b-c”, “0-a-b”, “a-b”, “a-b(c)”`.

2) The lists relative to the syntaxes are then the following (“|” indicates separation between recognized segments):

Syntax: a-b-c
```
Various | Animals | The house of the rising sun
Various | The Mindbenders | A groovy kind of love
Various | Hollies | The Air That I Breathe
Various | The Beatles | Ain't she sweet
Various | Bee Gees | Massachusetts
Various | The Moody Blues | Nights in white satin
```

Syntax: 0-a-b
```
01 | The Beatles | Doctor Robert
05 | The Beatles | Sgt Pepper’s Lonely Hearts
```

Syntax: a-b
```
Simon and Garfunkel | The Sound of Silence
Bee Gees | Saturday Night Fever
Beastie Boys | Song for Junior
Beach Boys | Good Vibrations
```

Syntax: a-b(c)
```
Simon and Garfunkel | El Condor Pasa (If I Could)
```

Each of these three sub clusters is now treated using the implementation of the heuristics as described below.
In the next sections, we consider each sub cluster as an array. The lines of the array match the lines of the sub cluster, and the columns of the array match the segments in each line of the sub cluster.

4.3 Management of Identifiers

In order to take into account differences in the spelling of Proper names (artists) and title names, we implement retrieval mechanisms based on a canonical representation of identifiers. This representation is computed so that different spellings of a given identifier yield the same representation.
The principle is to build a unique String composed only of the significant characters of a given identifier, removing blanks, spaces, and non-standard characters.

Additionally, there is a specific provision for managing artist names: artist names may have several attributes such as “firstName”, or “group prefix” (e.g. “The” or “Les” in French). These attributes are specified in a lazy mode, that is as they are encountered.
For instance, the first time we encounter the artist spelled as “McCartney, Paul”, we create an entry in the artist directory, with a canonical representation being “mccartney”, a first name being “paul”.
When we encounter another occurrence of McCartney, but with a different ordering or spelling, such as “McCartney” or “Paul McCartney”, we are able to retrieve the previously entered occurrence by trying several all the possible combinations.

Lastly, this specific procedure is augmented with a fuzzy matching algorithm to take into account possible misspelling and errors. This procedure is not discussed here for reasons of space.

4.4 Implementation of the heuristics

We describe here how we implement and prioritise the different heuristics to infer the artist and title information from a given sub cluster in which all titles share the same syntax. We do not describe the whole analyser here, but only highlight its main structure.

4.4.1 Case 1, implicit information

If the sub cluster contains only one segment, the only hypothesis we can make is that 1) the segment denotes the title name, and 2) the artist information is contained in the super cluster (super directory usually). For instance, if the corpus is a directory from a personal database of music file names, the artist name can be the name of the directory containing the music files. This is the case with the 2 “Stan Getz” files in Figure 2 for instance.

4.4.2 General Case

As illustrated in Section 2.2, about 93% of the play lists analysed from CDDB have at least two segments. We therefore assume that these segments contain at least both the title name and the artist name. The problem is now to determine which segment is the artist name, which one the title name, and which ones are useless groups of words such as constant terms, dates, etc.

Here is the ordering of the heuristics to identify properly the artist and title information.

1) Constant term heuristics,
2) Artist names heuristics,
3) Title name heuristics.

4.4.2.1 Constant Terms Heuristics

This heuristics is applied only if the syntax of the sub cluster considered contains at least two segments.
We first check if the array contains any constant terms in a whole column. If a column contains the same constant term, there are two possible interpretations:
- The array contains two columns: the column containing the constant terms is assumed to be the artist name column.
- The array contains more than two columns: we must check if the constant term belongs to a list of known constant terms as illustrated in section 3.3. The list of well-known constant terms we use has been retrieved from our CDDB database. We cannot determine whether or not a constant term is an artist name if it does not belong to our list. If the constant term belongs to our list of constant terms, we will not take into account the column relative to this constant term anymore in the title identification and consider that the artist and title names are the remaining columns of the array.

4.4.2.2 Artist Names Heuristics

To determine if a column is an artist name, we consider the following information in the following order:

1) Number of comas.

One heuristic is to consider that the column containing the most commas in its strings is the artist names column. Indeed, even if the percentage of cases where the artist name is written with a comma (ex: “Beatles, the”, “Mc Cartney, paul”) is not very high, this is a first way to retrieve the artist name.

2) Known artists

Then, if the artist column has not been found, we propose to take into account the artists already known by the system. If a known artist is found in a column, this is the artist column, following our local consistency hypothesis (all the artist names are in the same column).

3) Number of different words

Once the elimination of columns containing useless terms has been performed and if there are only two columns left in the array considered, we count the number of different words in all the valid columns of the array. If the number is smaller in one of the columns, we assume this column represents the artist names.

4.4.2.3 Title Name Heuristics

At this step of the identification, the column of the array containing the title names may be inferred in most of the cases by elimination since there is most often only one column remaining.
However, if we have more than one column, we apply again the heuristics about the number of words: if the number of different words is greater in one of the columns, we considered it as the title names column.

5 Experimentations

Our system has been tested and validated on our three databases. To validate the system, we made about 1500 random experiments, by drawing a random title, and checking manually whether the parse was correct or not. 95 % of the cases where correctly analysed. We assume the most frequent cases have been encountered.

Figure 10. The interface of FNI.

The incorrect cases are most often non interpretable file names. For instance, as illustrated in Figure 10, “Various - Toots Thielemans - Jane's Theme - 05” has too many segments. The “05” is duly recognized as a constant term, but the system cannot determine which segment refers to the title name and which one refers to the artist name. In this case, even a human could not infer the right syntax,

unless he/she would know the track listings and albums names of Toots Thielmans.
A few cases were not correctly analysed because the syntax exceptionally did not match our heuristics. Example: “Johnny Lee Hooker - Boom, Boom.mp3”. The artist name has more words than the title name, and the title name contains a comma. However our system allows to correct manually the wrong file names (see Figure 10). Additionally, the list of “known artists” is updated automatically, so mistakes are only done once.
FNI is integrated in *Personal Radio*, a working EMD system that has already been tested on over 100 users. More tests are being conducted within the Cuidado European project (Pachet, 2001b).

6 Conclusion

We described a method for parsing music file names without knowing their syntax *a priori*. The method is based on a set of justified heuristics which are validated by a prior analysis of a large corpora of “natural” file names, and by a working system integrated in a large scale EMD project. The success of the approach lies mainly in the local consistency hypothesis, which states that syntaxes are usually consistent within related groups of music files. This hypothesis allows to solve a number of ambiguity by making choices based on statistical properties of file clusters rather than on individual files. Extensions of the approach for handling other types of music (e.g. Classical) or non-Western filenames are under study, and may require different sets of heuristics, but we believe the approach in general is still valid. Lastly, we plan to integrate a learning module to FNI that is able to learn automatically new syntaxes from errors, in the spirit of (Petasis et al., 2001).

7 References

Beeferman, D. Berger, A. Lafferty, J. (1999) Statistical Models for Text Segmentation, Machine Learning, 34, 1-3, Feb. 1999.

Hacker, Scott (2000) *MP3, the definitive guide,* O’Reilly.

Maurel, D. Piton, O. Eggert, E. (2001) Automatic Processing of Proper Nouns Vol. 41 N.3, February 2001.

Pachet, F. (2001a) Content management for Electronic Music Distribution: The Real Issues, submitted to Communications of the ACM, 2001.

Pachet, F. (2001b) Metadata for music and sounds: The Cuidado Project, Content-Based Multimedia Indexing Workshop, Brescia (It).

Petasis, G. Vichot, F. Wolinski, F. Paliouras, G. Karkaletsis, V. Spyropoulos, C. (2001) Using Machine Learning to Maintain Rule-based Named-Entity Recognition and Classification Systems, Association for Computational Linguistics, ACL.