Chapter 8
Applications

This chapter presents two implemented prototypes that are related with the main topics presented in the book; music discovery and recommendation. The first system, named, *Searchsounds*, is a music search engine based on text keyword searches, as well as a *more like this* button, that allows users to discover music by means of audio similarity. Thus, *Searchsounds* allows users to dig into the Long Tail, by providing music discovery using audio content-based similarity. The second system, named *FOAFing the Music*, is a music recommender system that focuses on the Long Tail of popularity, promoting unknown artists. The system also provides related information about the recommended artists, using information available on the web gathered from music related RSS feeds.

The main difference between the two prototypes is that *Searchsounds* is a non-personalised music search engine, whilst *FOAFing the Music* takes into account the user profile and the listening habits to provide personalised recommendations.

8.1 Searchsounds: Music Discovery in the Long Tail

Searchsounds, is a web-based music search engine that allows users to discover music using content-based similarity. Section 8.1.1 introduces the motivations and background of the system implemented. In Sec. 8.1.3 we present the architecture of the system. Finally, the last section summaries the work done and outlines the remaining work regarding the functionality of the system.

8.1.1 Motivation

Nowadays, the increasing amount of available music in the World Wide Web makes very difficult, to the user, to find music she would like to listen to. To overcome this problem, there are some audio search engines that can fit the user's needs.

Ò. Celma, *Music Recommendation and Discovery*,
DOI 10.1007/978-3-642-13287-2_8,

Some of the current existing search engines are, nevertheless, not fully exploited because their companies would have to deal with copyright infringing material. As general search engines, music search engines have a crucial component: an audio crawler, that scans the web for audio files, and also gathers related information about files [1].

8.1.1.1 Syndication of Web Content

During the last years, syndication of web content—a section of a website made available for other sites to use—has become a common practice for websites. This originated with news and weblog sites, but nowadays is increasingly used to syndicate any kind of information. Since the beginning of 2003, a special type of weblog, named audio weblogs (or MP3 blogs), has become very popular. These blogs make music titles available for download. The posted music is explained by the blog author, and usually it has links that allow users to buy the complete album or work. Sometimes, the music is hard to find or has not been issued in many years, and many MP3 blogs link strictly to music that is authorised for free distribution. In other cases, MP3 blogs include a disclaimer stating that they are willing to remove music if the copyright owner objects. Anyway, this source of semi-structured information is a jewel for web crawlers, as it contains the user's object of desire—the music—and some textual information that is referring to the audio file.

The file format used to syndicate web content is XML. Web syndication is based on the RSS family and Atom formats. The RSS abbreviation is used to refer to the following standards: Really Simple Syndication (RSS 2.0), Rich Site Summary (RSS 0.91 and 1.0) or RDF Site Summary (1.0).

Of special interest are the feeds that syndicate multimedia content. These feeds publish audiovisual information that is available on the net. An interesting example is the Media RSS (mRSS) specification,[1] lead by *Yahoo!* and the multimedia RSS community. mRSS allows bloggers to syndicating multimedia files (audio, video, image) in RSS feeds, and adds several enhancements to RSS enclosures. Although mRSS is not yet widely used on the net, some websites syndicate their multimedia content following the specification. These feeds contain textual information, plus a link to the actual audiovisual file. As an example, Listing 8.1 shows a partial RSS feed.[2]

```
<rss version="2.0"
xml:base="http://www.ourmedia.org"
xmlns:media="http://search.yahoo.com/mrss"
xmlns:dc="http://purl.org/dc/elements/1.1/"
>
<channel>
 <title>Example of a mRSS feed</title>
 <link>http://www.ourmedia.org/user/45801</link>
```

[1] http://search.yahoo.com/mrss/

[2] Adapted from a real example published in *OurMedia* website. http://www.ourmedia.org

```
 <description>
  Recently published media items from Ourmedia.org
 </description>
 <language>en</language>
 <item>
  <title>Fanky beats</title>
  <link>http://www.ourmedia.org/node/...</link>
  <description>Rock music with a funky beat and electric lead
      guitar riffs (...)</description>
  <pubDate>Mon, 17 Apr 2007 01:35:49 -0500</pubDate>
  <dc:creator>John Brettbutter</dc:creator>
  <category domain="urn:ourmedia:term:35">
   Alternative Rock
  </category>
  <category domain="urn:ourmedia:term:582">funk</category>
  <category domain="urn:ourmedia:term:727">guitar</category>
  <enclosure url="http://archive.org/.../file.mp3"
   length="3234212" type="application/octet-stream" />
 </item>
 <item>
  <title>Another item</title>
  ...
 </item>
</channel>
</rss>
```

Listing 8.1 Example of a media RSS feed.

The example shows an item with all its information: the title of the item, the description, the publication date, the editor of the entry, and a set of categories (similar to tags, but controlled from a given taxonomy). *Searchsounds* mines this information in order to retrieve relevant audio files based on keywords.

8.1.2 Goals

The main goal of the system is to allow users to discover unknown music. For this reason, *Searchsounds* mines music related information available in MP3-weblogs, and attaches textual information to the audio files. This way, users can search and retrieve music related to the query, as well as music that sounds similar to the retrieved audio files. This exploration mode allows users to discover music—related to his original (keyword based) query—that would be more difficult to discover using only textual queries.

Figure 8.1 shows the relationship between the music information plane (see Sec. 3.3), and the information that *Searchsounds* uses.

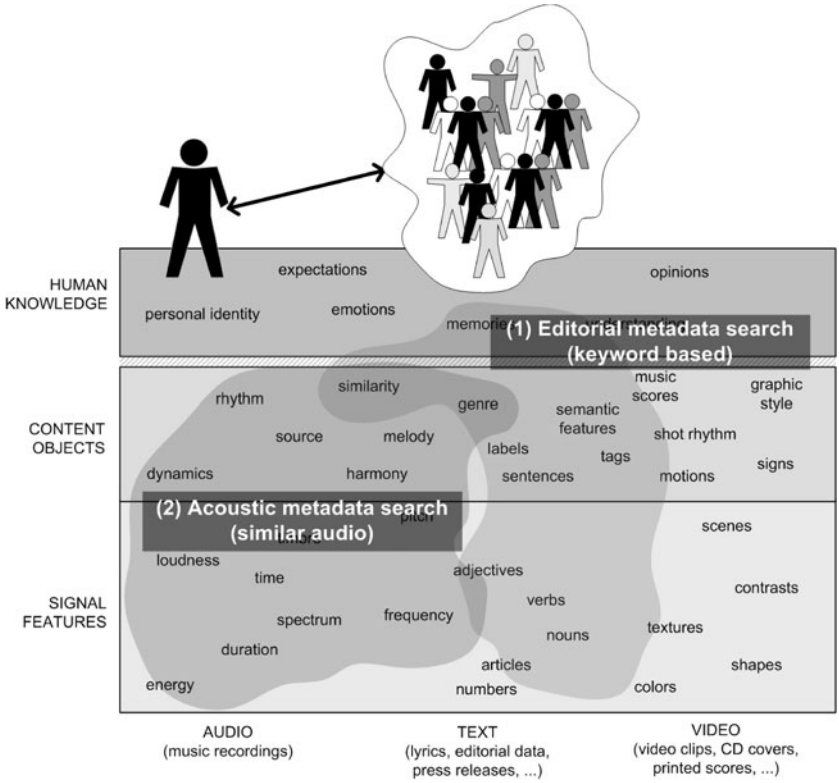

Fig. 8.1 *Searchsounds* makes use of editorial, cultural and acoustic metadata. The system retrieves (1) audio files from a keyword query, as well as (2) a list of (content-based) similar titles.

8.1.3 System Overview

Searchsounds exploits and mines all the music related information available from MP3-weblogs. The system gathers editorial, cultural, and acoustic information from the crawled audio files. The input of the system is a query composed by text keywords. From these keywords, the system is able to retrieve a list of audio files related with the query. Each audio file provides a link to the original weblog, and a list of similar titles. This similarity is computed using content-based audio description. Thus, from the results of a keyword query, a user can discover related music by navigating onto the audio similarity plane. It is worth to mention that there is no user profiling or any kind of user representation stored in the system. This is a limitation, as the system does not make any personalised recommendations. However, this limitation is solved in the next prototype (explained in Sec. 8.2). The main components of the system are the audio crawler and the audio retrieval system. Figure 8.2 depicts the architecture of the system.

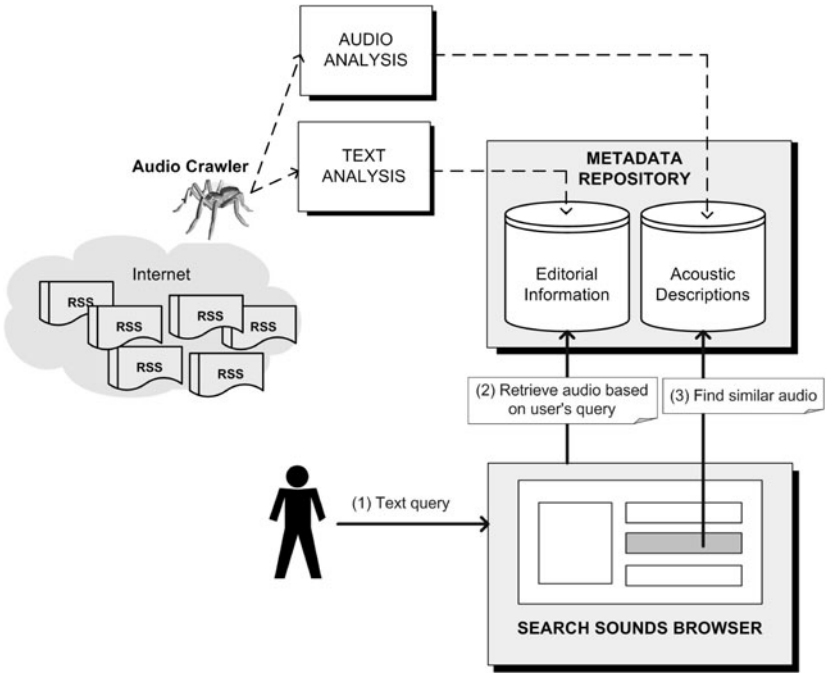

Fig. 8.2 *Searchsounds* architecture. The main components are the audio crawler, and the audio retrieval system.

8.1.3.1 Audio Crawler

The system has an audio spider module that crawls the web. All the gathered information is stored into a relational database. The audio crawler starts the process from a manually selected list of RSS links (that point to MP3-blogs). Each RSS file contains a list of entries (or *items*) that link to audio files. The crawler seeks for new incoming items—using the *pubDate* item value and comparing with the latest entry in the database—and stores the new information into the database. Thus, the audio crawler system has an historic information of all the items that appeared in a feed.

From the previous RSS example (see Example 8.1, presented in Section 8.1.1.1), the audio crawler stores the *title*, the content of the *description*, the assigned terms from the taxonomy (*category* tags), and the link to the audio file (extracted from the *enclosure url* attribute).

8.1.3.2 Audio Retrieval System

The logical view of a crawled feed item can be described by the bag-of-words approach: a document is represented as a number of unique words, with a weight (in

our case, the tf/idf function) assigned to each word [2]. Special weights are assigned to the music related terms, as well as the metadata (e.g. ID3 tags) extracted from the audio file. Similar to our approach, [3] presents a proposal of modifying the weights of the terms pertaining to the musical domain.

Moreover, basic natural language processing methods are applied to reduce the size of the item description (elimination of stopwords, and apply Porter's stemming algorithm [4]). The information retrieval (IR) model used is the classic vector model approach, where a given document is represented as a vector in a multidimensional space of words (each word of the vocabulary is a coordinate in the space).

The similarity function, $sim(d_j, q)$, between a query (q) and a document (d_j) is based on the cosine similarity, using $TF\text{-}IDF$ weighting function (already presented in Sec. 2.5.4). Our approach is well suited not only for querying via artists' or songs' names, but for more complex keyword queries such as: *funky guitar riffs* or *traditional Irish tunes*. The retrieval system outputs the documents (i.e. feed entries) that are relevant to the user's query, ranked by the similarity function. Figure 8.3 depicts the retrieved audio files for *traditional Irish music* query.

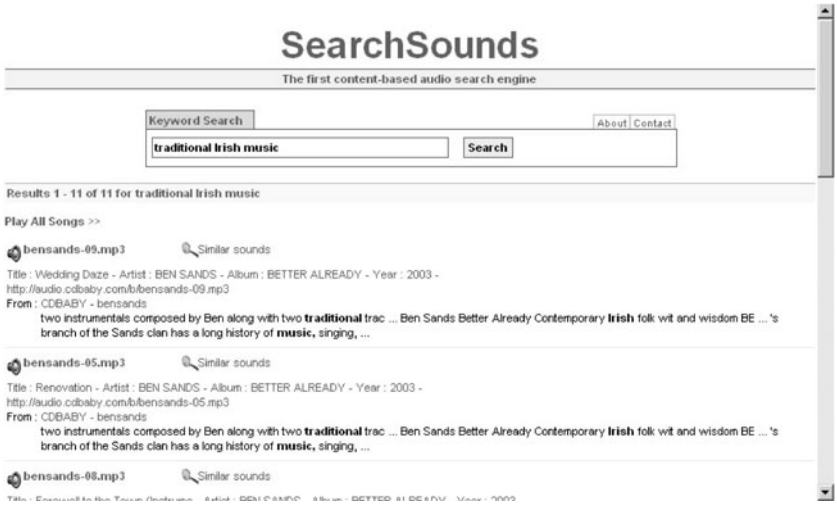

Fig. 8.3 Screenshot of the *Searchsounds* application, showing the first 10 results from *traditional Irish music* query.

Based on the results obtained from the user's textual query, the system allows users to find similar titles using content-based audio similarity. Each link to an audio file has a *Find similar* button that retrieves the most similar audio files, based on a set of low and mid-level audio descriptors. These descriptors are extracted from the audio and represent properties such as: rhythm, harmony, timbre and instrumentation, intensity, structure and complexity [5].

This exploration via browsing allows users to discover music—related to his original (keyword based) query—that would be more difficult to discover by using textual queries only. There is an analogy between this type of navigation and, for example, Google's "find web pages that are similar to a given HTML page". In our case, similarity among items are based on audio similarity, whereas Google approach is based on the textual content of the HTML page. Still, both browsing approaches are based on the *content* analysis of the retrieved object.

8.1.4 Summary

We developed a web-based audio crawler that focuses on MP3-weblogs. Out of the crawling process, each feed item is represented as a text document, containing the content of the item, as well as the links to the audio files. Then, classic text retrieval system outputs relevant feed items related to the user's query. Furthermore, a content-based navigation allows users to browse through the retrieved items and discover new music and artists using audio similarity.

Ongoing work includes the automatic extraction of music related tags (i.e. *guitar*, *rock*, 1970s) from the text, as well as applying autotagging to incoming audio files; using audio content-based similarity [6]. We also plan to add relevance feedback to tune the system and get more accurate results, specially for the content-based similarity.

The system is available at `http://www.searchsounds.net`.

8.2 FOAFing the Music: Music Recommendation in the Long Tail

Now we present the second of the two prototypes developed. It is a music recommender system, named *FOAFing the Music*, that allows users to discover a wide range of music located along the Long Tail. The system exploits music related information that is being syndicated (as RSS feeds) on thousands of websites. Using the crawled information, the system is able to filter it and recommend it to the user, according to her profile and listening habits.

8.2.1 Motivation

The World Wide Web has become the host and distribution channel for a broad variety of digital multimedia assets. Although the Internet infrastructure allows simple straightforward acquisition, the value of these resources lacks powerful content management, retrieval and visualisation tools. Music content is no exception: although there is a sizeable amount of text-based information related to music (album

reviews, artist biographies, etc.) this information is hardly ever associated with the objects it refers to, that being the music files themselves (MIDI or audio). Moreover, music is an important vehicle for communicating to other people something relevant about our personality, history, etc.

There is a clear interest in the Semantic Web field in creating a Web of machine-readable homepages describing people, the links among them, and the things they create and do. The Friend of a Friend (*Friend Of A Friend*) project[3] provides conventions and a language to describe homepage-like content and social networks. The Friend of a Friend vocabulary provides properties and classes for describing common features of people and their social networks. Friend of a Friend is based on the Resource Description Framework (RDF[4]) vocabulary.

We foresee that with a complete user's Friend of a Friend profile, our system would get a better representation of the user's musical needs. On the other hand, the RSS vocabulary[5] allows systems one to syndicate Web content on the Internet. Syndicated content includes data such as news, event listings, headlines, project updates, as well as music related information, such as new music releases, album reviews, podcast sessions, and upcoming gigs.

To our knowledge, nowadays it does not exist any system that recommends items to a user, based on her Friend of a Friend profile. Yet, it is worth to mention the *FilmTrust* system.[6] It is a part of a research study aimed to understanding how social preferences might help web sites to present information in a more useful way [7]. The system collects user reviews and ratings about movies, and holds them into the user's Friend of a Friend profile [8].

8.2.2 Goals

The main goal of the *FOAFing the Music* system is to recommend, to discover and to explore music content; based on user profiling (via Friend of a Friend descriptions), context based information (extracted from music related RSS feeds), and content based descriptions (automatically extracted from the audio itself). All of that being based on a common ontology that describes the musical domain.

Figure 8.4 shows the relationship between the music information plane, and the different sources of metadata that the system exploits. Compared to the first prototype (*Searchsounds*), *Foafing the Music* holds a user profile representation, based on the Friend of a Friend initiative (already presented in Sec. 3.2). A Friend of a Friend user profile allows to filter music related information according to user's preferences.

[3] http://www.foaf-project.org

[4] http://www.w3.org/RDF

[5] http://web.resource.org/rss/1.0/

[6] http://trust.mindswap.org/FilmTrust

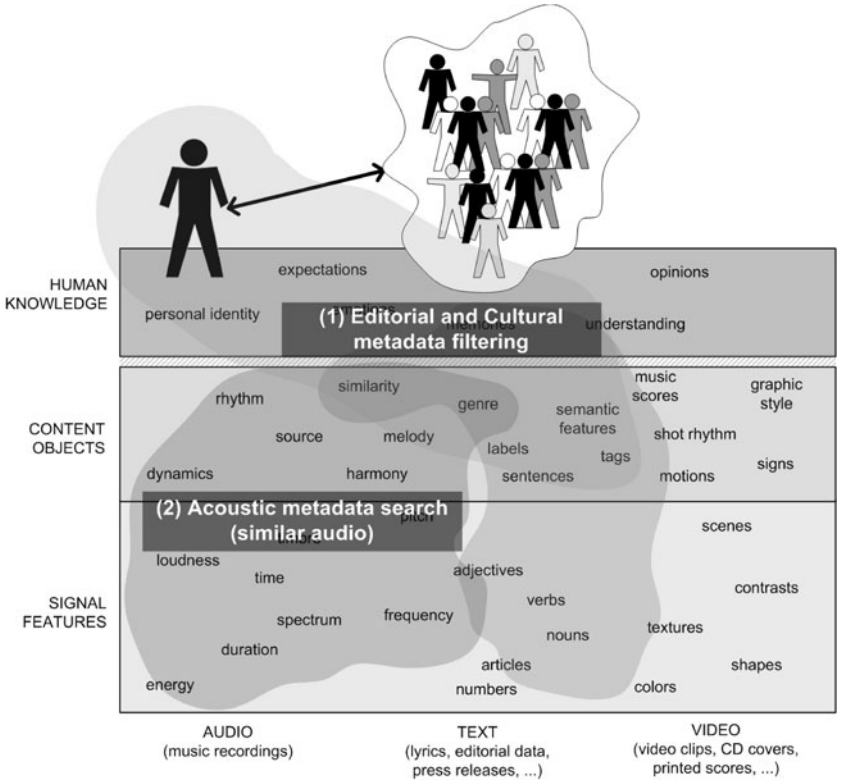

Fig. 8.4 *FOAFing the Music* and the music information plane.

8.2.3 System Overview

The overview of the *Foafing the Music* system is depicted in Fig. 8.5. The system is divided in two main components, that is (i) how to gather data from external third party sources (presented in Sec. 8.2.3.1), and (ii) how to recommend music to the user based on the crawled data, and the semantic description of the music titles (Sec. 8.2.3.3).

8.2.3.1 Gathering Music Related Information

Personalised services can raise privacy concerns due to the acquisition, storage and application of sensitive personal information [9]. In our system, information about the user is not stored in the system in any way. Instead, the system has only a link pointing to the user's Friend of a Friend profile (often a link to a *Livejournal* ac-

count). Thus, the sensitivity of this data is up to the user, not to the system. Users' profiles in *Foafing the Music* are distributed over the net.

Regarding music related information, our system exploits the mashup approach. The system uses a set of public available APIs and web services sourced from third party websites. This information can come in any of the different RSS formats (v2.0, v1.0, v0.92 and Yahoo! Media RSS), as well as in the Atom format. Thus, the system has to deal with syntactically and structurally heterogeneous data. Moreover, the system keeps track of all the new items that are published in the feeds, and stores the new incoming data in a historic relational database. Input data of the system is based on the following information sources:

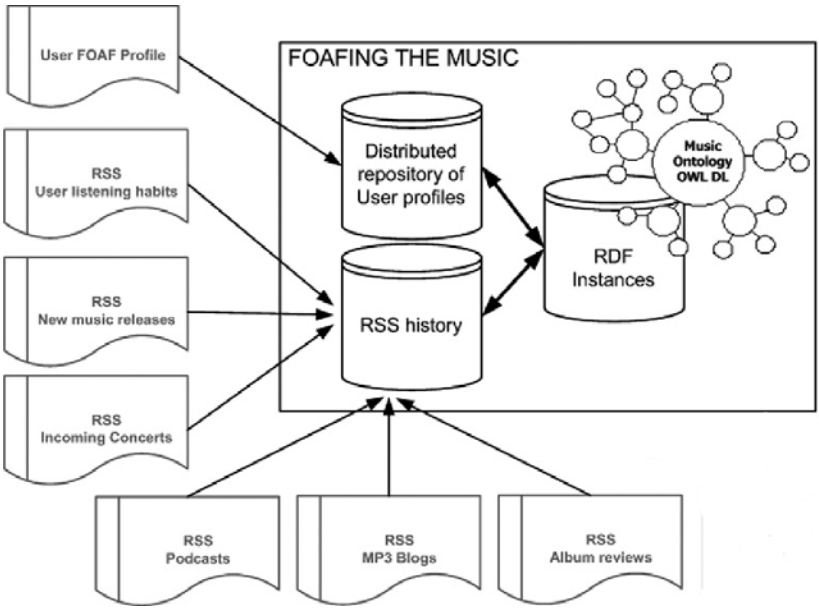

Fig. 8.5 Architecture of the *Foafing the Music* system.

- *User listening habits*. To keep track of the user's listening habits, the system uses the services provided by *last.fm*. This system offers a list of RSS feeds that provide the most recent tracks a user has played. Each item feed includes the artist name, the song title, and a timestamp—indicating when the user has listened to the track.
- *New music releases*. The system uses a set of RSS feeds that gathers new music releases from *iTunes*, *Amazon*, *Yahoo! Shopping* and *Rhapsody*.
- *Upcoming concerts*. The system uses a set of RSS feeds that syndicates music related events. The websites are: *Eventful.com*, and *Upcoming.org*. Once the system has gathered the new items, it queries the Google Maps API to get the

geographic location of the venues, so it can be filtered according to the user's location.
- *Podcast sessions*. The system gathers information from a list of RSS feeds that publish podcast sessions.
- *MP3 Blogs*. The system gathers information from a list of MP3 blogs that talk about artists and new music releases.
- *Album reviews*. Information about album reviews are crawled from the RSS feeds published by *Rateyourmusic.com*, *Pitchforkmedia.com*, online magazines *Rolling Stone*,[7] *BBC*,[8] *New York Times*,[9] and *75 or less records*.[10]

Table 8.1 shows some basic statistics of the data that has been gathered since mid April, 2005 until the first week of March, 2010. These numbers show that the system has to deal with daily incoming data.

Source	# RSS seed feeds	# Items stored
New releases	44	1,283,640
MP3 blogs	127	991,997
Podcasts	833	288,992
Album reviews	18	206,265
Upcoming concerts	16	369,651

Table 8.1 Information gathered from music related RSS feeds is stored into a relational database. Based on the user's Friend of a Friend profile, the system filters this information, and presents the most relevant items according to her musical taste.

8.2.3.2 Music Ontologies

An ontology is an explicit and formal specification of a conceptualisation [10]. In general, an ontology describes formally a domain of discourse. The requeriments for Ontology languages are: a well-defined syntax, a formal semantics, and a reasoning support that checks the consistency of the ontology, checks for unintended relationships between classes, and automatically classifies instances in classes.

The Web Ontology Language (OWL[11]) has a richer vocabulary description language for describing properties and classes than RDF Schema (RDFS[12]). OWL has relations between classes, cardinality, equality, characteristics of properties and enumerated classes. The OWL language is build on top of RDF and RDFS, and uses RDF/XML syntax. OWL documents are, then, RDF documents.

[7] http://www.rollingstone.com/
[8] http://www.bbc.co.uk/
[9] http://www.nytimes.com/
[10] http://www.75orless.com/
[11] http://www.w3.org/TR/owl-guide/
[12] http://www.w3.org/TR/rdf-schema/

On the other hand, we have defined a simple music recommendation OWL DL ontology[13] that describes some basic properties of the artists and music titles, as well as some descriptors automatically extracted from the audio files (e.g. tonality, rhythm, moods, music intensity, etc.). In [11] we propose a way to map our ontology and the *Musicbrainz* ontology, onto the MPEG-7 standard, which acts as an upper-ontology for multimedia description. This way we can link our dataset with the *Musicbrainz* information in a straightforward manner.

A focused web crawler has been implemented to add instances to our music ontology. The crawler extracts metadata of artists and songs, and the relationships between artists (such as: "related with", "influenced by", "followers of", etc.), and converts it to RDF/XML notation. The seed sites to start the crawling process are music metadata providers, such as *MP3.com*, *Yahoo! Music*, and *RockDetector*, as well as independent music labels (*Magnatune*, *CDBaby*, *Garageband*, etc.).

Based on our lightweight music recommendation ontology, listing 8.2 shows the RDF/XML description of an artist from *GarageBand*.

```
<rdf:Description rdf:about="http://www.garageband.com/artist/
    randycoleman">
 <rdf:type rdf:resource="{\&}music;Artist"/>
 <foaf:name>Randy Coleman</foaf:name>
 <music:decade>1990</music:decade>
 <music:decade>2000</music:decade>
 <music:genre>Pop</music:genre>
 <foaf:based_near
   rdf:resource="http://sws.geonames.org/5368361/"/>
 <music:influencedBy
   rdf:resource="http://www.coldplay.com"/>
 <music:influencedBy
   rdf:resource="http://www.jeffbuckley.com"/>
 <music:influencedBy
   rdf:resource="http://www.radiohead.com"/>
</rdf:Description>
```

Listing 8.2 RDF example of an artist individual

Listing 8.3 shows the description of an individual track of the previous artist, including basic editorial metadata, and some features extracted automatically from the audio file.

```
<rdf:Description rdf:about="http://www.garageband.com/song?|pe1|
    S8LTM0LdsaSkaFeyYG0">
 <rdf:type rdf:resource="{\&}music;Track"/>
 <music:title>Last Salutation</music:title>
 <music:playedBy rd:resource="http://www.garageband.com/artist/
     randycoleman"/>
 <music:duration>247</music:duration>
 <music:intensity>Energetic</music:intensity>
 <music:key>D</music:key>
 <music:keyMode>Major</music:keyMode>
 <music:tonalness>0.84</music:tonalness>
 <music:tempo>72</music:tempo>
</rdf:Description>
```

[13] http://foafing-the-music.iua.upf.edu/music-ontology#

Listing 8.3 Example of a track individual

These individuals are used in the recommendation process, to retrieve artists and songs related with the user's musical taste.

8.2.3.3 Providing Music Recommendations

This section explains the music recommendation process, based on all the information that has continuously been gathered from the RSS feeds and the crawler. Music recommendations, in the *Foafing the Music* system, are generated according to the following steps:

1. Get music related information from user's Friend of a Friend interests, and listening habits from *last.fm*,
2. Detect artists and bands,
3. Compute similar artists, and
4. Rate the results by relevance, according to the user's profile.

To gather music related information from a Friend of a Friend profile, the system extracts the information from the FOAF interest property (if `dc:title` is given then it gets its value, otherwise it gathers the text from the `<title>` tag of the HTML resource).

```
<foaf:interest
  rdf:resource="http://www.tylaandthedogsdamour.com/"
  dc:title="The␣Dogs␣d'Amour" />
```

Listing 8.4 Example of a Friend of a Friend interest with a given `dc:title`.

The system can also extract information from a user's Friend of a Friend interest that includes the artist description based on the general Music Ontology [12].

The following example presents a way to express interest in an artist, by means of the general Music Ontology.

```
<foaf:interest>
  <mo:MusicArtist rdf:about='http://musicbrainz.org/artist/12
      d432a3-...-d20751880764'>
    <mo:discogs rdf:resource='http://www.discogs.com/artist/Yann+
        Tiersen'/>
    <foaf:img rdf:resource='http://ec2.images-amazon.com/images/P
        /B000852GIQ...Z_.jpg'/>
    <foaf:homepage rdf:resource='http://www.yanntiersen.com/'/>
    <foaf:name>Yann Tiersen</foaf:name>
    <mo:wikipedia rdf:resource='http://en.wikipedia.org/wiki/
        Yann_Tiersen'/>
  </mo:MusicArtist>
</foaf:interest>
```

Listing 8.5 FOAF example of an artist description that a user is interested in.

Based on the music related information gathered from the user's profile and listening habits, the system detects the artists and bands that the user is interested in,

by doing a SPARQL query to the artist RDF repository. Once the user's artists have been detected, artist similarity is computed. This process is achieved by exploiting the RDF graph of artists' relationships (e.g. *influenced by*, *followers of*, *worked with*, etc.), as shown in Listing 8.2.

The system offers two ways of recommending music information. On the one hand, *static* recommendations are based on the favourite artists encountered in the Friend of a Friend profile. We assume that a Friend of a Friend profile would be rarely manually updated or modified. On the other hand, *dynamic* recommendations are based on user's listening habits, which are updated much more often than the user's profile. Following this approach a user can discover a wide range of new music and artists on a daily basis.

Once the recommended artists have been computed, *Foafing the Music* filters music related information coming from the gathered music information (see Sec. 8.2.3.1) to:

- Get new music releases from iTunes, Amazon, Yahoo Shopping, etc.
- Download (or stream) audio from MP3-blogs and Podcast sessions,
- Create, automatically, XSPF[14] playlists based on audio similarity,
- View upcoming gigs happening near to the user's location, and
- Read album reviews.

Syndication of the website content is done via an RSS 1.0 feed. For most of previous functionalities, there is a feed subscription option to get the results.

8.2.3.4 Usage Data

Since its inception in August 2005, the system has an average of 60 daily unique accesses, from more than 5,000 registered users, including casual users that try the *demo* option. More than half of the users automatically created an account using an external Friend of a Friend profile (most of the times, around 70%, the profile came from their *Livejournal* Friend of a Friend account). Also, more than 65% of the users add her *last.fm* account, so we can use their listening habits from *last.fm*. Figure 8.6 shows the number of logins over time, since August 2005 till July 2008. The peaks are clearly correlated with related news about the project (e.g. local TV and radio interviews, and reviews on the web).

8.2.4 Summary

We have proposed a system that filters music related information, based on a given user's Friend of a Friend profile and her listening habits. A system based on Friend of a Friend profiles and user's listening habits allows the system to "understand" a

[14] http://www.xspf.org/. XSPF is a playlist format based on XML syntax

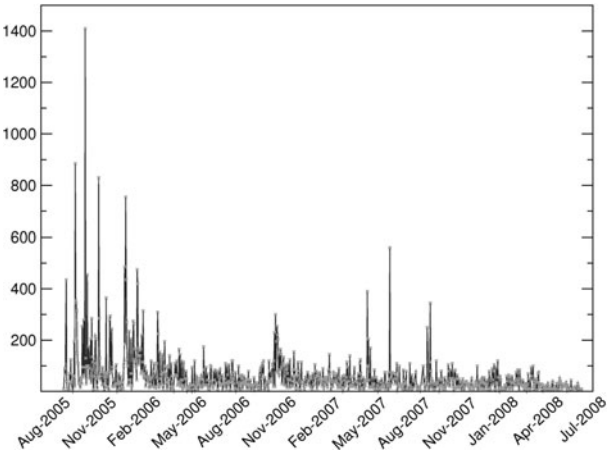

Fig. 8.6 Daily accesses to *Foafing the Music*. The system has an average of 60 daily unique accesses, from more than 4,000 registered users and also casual users that try the *demo* option.

user in two complementary ways; psychological factors—personality, demographic preferences, social relationships—and explicit musical preferences. In the music field, we expect that filtering information about new music releases, artists' interviews, album reviews, and so on, can improve user satisfaction as it provides the context and needed information to backup the system's recommendations.

Describing music assets is a crucial task for a music recommender system. The *success* of a music recommender can depend on the accuracy and level of detail of the musical objects, and its links within a user profile. Furthermore, we formalise into an ontology the basic musical concepts involved in the recommendation process. Linking these musical objects with the user profile eases the recommendation process.

Furthermore, high–level musical descriptors can increase the accuracy of content retrieval, as well as provide better personalised recommendations. Thus, going one step beyond, it would be desirable to combine mid–level acoustic features with as much editorial and cultural metadata as possible. From this combination, more sophisticated inferences and semantic rules would be possible. These rules could derive hidden high–level metadata that could be easily understood by the end-user, also enhancing their profiles. Since the existence of the general Music Ontology (MO) [12], we foresee that linking our recommendation ontology with it, as well as using all the linked information available in the Web of Data,[15] we can improve our recommender, becoming a truly semantically-enhanced music recommender.

Foafing the Music is available at `http://foafing-the-music.iua.upf.edu`.

[15] See `http://linkeddata.org/`.

References

1. I. Knopke, "Arooooga: An audio search engine for the world wide web," in *Proceedings of 5th International Conference on Music Information Retrieval*, (Barcelona, Spain), 2004.
2. R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA: Addison-Wesley, 1st edn., 1999.
3. S. Vembu and S. Baumann, "A self-organizing map based knowledge discovery for music recommendation systems," in *Proceedings of the 2nd International Symposium on Computer Music Modeling and Retrieval*, (Esbjerg, Denmark), 2004.
4. M. F. Porter, "An algorithm for suffix stripping," *Program,* vol. 14, pp. 130–137, 1980.
5. P. Cano, M. Koppenberger, and N. Wack, "An industrial-strength content-based music recommendation system," in *Proceedings of 28th International ACM SIGIR Conference*, (Salvador, Brazil), 2005.
6. M. Sordo, C. Laurier, and O. Celma, "Annotating music collections how content-based similarity helps to propagate labels," in *Proceedings of the 8th International Conference on Music Information Retrieval*, (Vienna, Austria), 2007.
7. J. Golbeck and B. Parsia, "Trust network-based filtering of aggregated claims," *International Journal of Metadata, Semantics and Ontologies*, vol. 1, no. 1, 2005.
8. J. Golbeck, *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, College Park, MD, 2005.
9. E. Perik, B. de Ruyter, P. Markopoulos, and B. Eggen, "The sensitivities of user profile information in music recommender systems," in *Proceedings of Private, Security, Trust*, 2004.
10. T. R. Gruber, "Towards principles for the design of ontologies used for knowledge sharing," in *Formal Ontology in Conceptual Analysis and Knowledge Representation* (N. Guarino and R. Poli, eds.). Deventer, The Netherlands: Kluwer Academic Publishers, 1993.
11. R. Garcia and O. Celma, "Semantic integration and retrieval of multimedia metadata," in *Proceedings of 4th International Semantic Web Conference. Knowledge Markup and Semantic Annotation Workshop*, (Galway, Ireland), 2005.
12. Y. Raimond, S. A. Abdallah, M. Sandler, and F. Giasson, "The music ontology," in *Proceedings of the 8th International Conference on Music Information Retrieval*, (Vienna, Austria), 2007.