Foafing the Music: Bridging the semantic gap in music recommendation

Òscar Celma[1]

Music Technology Group, Universitat Pompeu Fabra, Barcelona, SPAIN
`http://mtg.upf.edu`

Abstract. In this paper we give an overview of the *Foafing the Music* system. The system uses the *Friend of a Friend* (FOAF) and *RDF Site Summary* (RSS) vocabularies for recommending music to a user, depending on the user's musical tastes and listening habits. Music information (new album releases, podcast sessions, audio from MP3 blogs, related artists' news and upcoming gigs) is gathered from thousands of RSS feeds.
The presented system provides music discovery by means of: user profiling (defined in the user's FOAF description), context based information (extracted from music related RSS feeds) and content based descriptions (extracted from the audio itself), based on a common ontology (OWL DL) that describes the music domain.
The system is available at: **http://foafing-the-music.iua.upf.edu**

1 Introduction

The World Wide Web has become the host and distribution channel of a broad variety of digital multimedia assets. Although the Internet infrastructure allows simple straightforward acquisition, the value of these resources lacks of powerful content management, retrieval and visualization tools. Music content is no exception: although there is a sizeable amount of text–based information about music (album reviews, artist biographies, etc.) this information is hardly associated to the objects they refer to, that is music music files (MIDI and/or audio). Moreover, music is an important vehicle for communicating other people something relevant about our personality, history, etc.

In the context of the Semantic Web, there is a clear interest to create a Web of machine-readable homepages describing people, the links among them, and the things they create and do. The FOAF (*Friend Of A Friend*) project[1] provides conventions and a language to describe homepage–like content and social networks. FOAF is based on the RDF/XML[2] vocabulary. We can foresee that with the user's FOAF profile, a system would get a better representation of the user's musical needs. On the other hand, the RSS vocabulary[3] allows to syndicate Web content on Internet. Syndicated content includes data such as news,

[1] http://www.foaf-project.org
[2] http://www.w3.org/RDF
[3] http://web.resource.org/rss/1.0/

events listings, headlines, project updates, as well as music related information, such as new music releases, album reviews, podcast sessions, upcoming gigs, etc.

2 Background

The main goal of a music recommendation system is to propose, to the end-user, interesting and unknown music artists (and their available tracks, if possible), based on user's musical taste. But musical taste and music preferences are affected by several factors, even demographic and personality traits. Then, the combination of music preferences and personal aspects —such as: age, gender, origin, occupation, musical education, etc.— could improve music recommendations [7]. Some of this information can be denoted using FOAF descriptions.

Moreover, a desirable property of a music recommendation system should be the ability of dynamically getting new music related information, as it should recommend new items to the user once in a while. In this sense, there is a lot of freely available (in terms of licensing) music on Internet, performed by "unknown" artists that can suit perfectly for new recommendations. Nowadays, music websites are noticing the user about new releases or artist's related news, mostly in the form of RSS feeds. For instance, iTunes Music Store[4] provides an RSS (version 2.0) feed generator[5], updated once a week, that publishes new releases of artists' albums. A music recommendation system should take advantage of these publishing services, as well as integrating them into the system, in order to filter and recommend new music to the user.

2.1 Collaborative filtering versus content based filtering

Collaborative filtering method consists of making use of feedback from users to improve the quality of recommended material presented to users. Obtaining feedback can be explicit or implicit. Explicit feedback comes in the form of user ratings or annotations, whereas implicit feedback can be extracted from user's habits. The main caveats of this approach are the following: the *cold-start* problem, the novelty detection problem, the item popularity bias, and the enormous amount of data (i.e users and items) needed to get some reasonable results [3]. Thus, this approach to recommend music can generate some "silly" (or obvious) answers. Anyway, there are some examples that succeed based on this approach. For instance, Last.fm[6] or Amazon [4] are good illustration systems.

On the other hand, content based filtering tries to extract useful information from the items data collection, that could be useful to represent user's musical taste. This approach solves the limitation of collaborative filtering as it can recommend new items (even before the system does not know anything about that item), by comparing the actual set of user's items and calculating a distance with some sort of similarity measure. In the music field, extracting musical semantics

[4] http://www.apple.com/itunes

[5] http://phobos.apple.com/WebObjects/MZSearch.woa/wo/0.1

[6] http://www.last.fm

from the raw audio and computing similarities between music pieces is a challenging problem. In [5], Pachet proposes a classification of musical metadata, and how this classification affects music content management, as well as the problems to face when elaborating a ground truth reference for music similarity (both in collaborative and content based filtering).

2.2 Related systems

Most of the current music recommenders are based on collaborative filtering approach. Examples of such systems are: Last.fm, MyStrands[7], MusicMobs[8], Goombah Emergent Music[9], iRate[10], and inDiscover[11]. The basic idea of a music recommender system based on collaborative filtering is:

1. To keep track of which artists (and songs) a user listens to —through iTunes, WinAmp, Amarok, XMMS, etc. plugins,
2. To search for other users with similar tastes, and
3. To recommend artists (or songs) to the user, according to these similar listeners' taste.

On the other hand, the most noticeable system using (manual) content based descriptions to recommend music is Pandora[12]. The main problem of the system is the scalability, because all the music annotation process is done manually.

Contrastingly, the main goal of the *Foafing the Music* system is to recommend, to discover and to explore music content; based on user profiling (via FOAF descriptions), context based information (extracted from music related RSS feeds), and content based descriptions (automatically extracted from the audio itself [1]). All of that being based on a common ontology that describes the musical domain. To our knowledge, nowadays it does not exist any system that recommends items to a user, based on FOAF profiles. Yet, there is the *FilmTrust* system[13]. 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. The system collects user reviews and ratings about movies, and holds them into the user's FOAF profile.

3 System overview

The overview of the system is depicted in Fig. 1. The next two sections explain the main components of the system, that is how to gather data from third party sources, and how to recommend music to the user based on crawled data, semantic description of music titles, and audio similarity.

[7] http://www.mystrands.com

[8] http://www.musicmobs.com

[9] http://goombah.emergentmusic.com/

[10] http://irate.sourceforge.net

[11] http://www.indiscover.net/

[12] http://www.pandora.com/

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

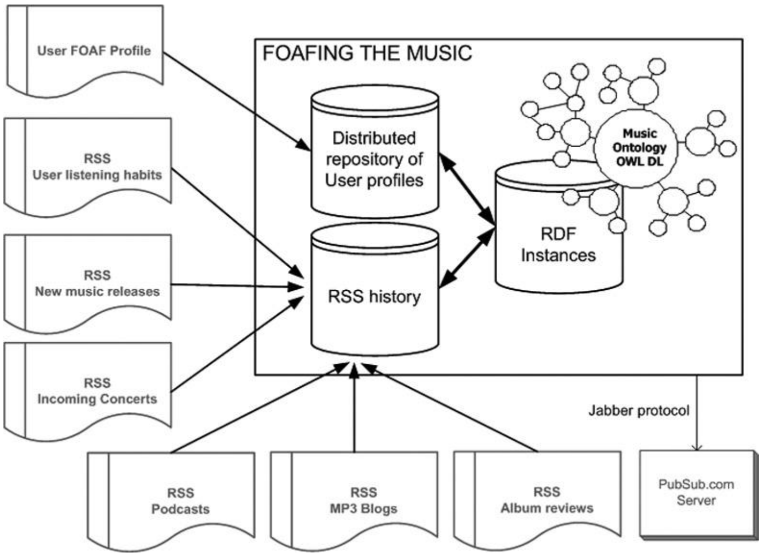

Fig. 1. Architecture of the *Foafing the Music* system.

3.1 Gathering music related information

Personalized services can raise privacy concerns, due to the acquisition, storage and application of sensitive personal information [6]. A novelty approach is used in our system: information about the users is not stored into the system in any way. Users' profiles are based on the FOAF initiative, and the system has only a link pointing to the user's FOAF URL. 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 family (v2.0, v1.0, v0.92 and mRSS), 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 into a historic relational database. Input data of the system is based on the following information sources:

- **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 web–based API —as well as a list of RSS feeds— that provide the most recent tracks a user has played. Each item feed includes, then, 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 that notifies new music releases. Next table shows the contribution of each RSS feed into the historic database of the system:

RSS Source	Percent
iTunes	45.67%
Amazon	42.33%
Oldies.com	2.92%
Yahoo Shopping	0.29%
Others	8.79%

- **Upcoming concerts**. The system uses a set of RSS feeds that syndicates music related events. The websites are: *Eventful.com*, *Upcoming.org*, *San Diego Reader*[14] and *Sub Pop* record label[15]. Once the system has gathered all the new items, it queries to the Google Maps API to get the geographic location of the venues.
- **Podcast sessions**. The system gathers information from a list of RSS feeds that publish podcasts sessions.
- **MP3 Blogs**. The system gathers information from a list of MP3 blogs that talk about artists and songs. Each item feed contains a list of links to the audio files.
- **Album reviews**. Information about album reviews are crawled from the RSS published by *Rateyourmusic.com*, *Pitchforkmedia.com*, *75 or less records*[16], and *Rolling Stone* online magazine[17].

RSS Source	# Seed feeds	# Items crawled per week	# Items stored
New releases	44	980	58,850
Concerts	14	470	28,112
Podcasts	830	575	34,535
MP3 blogs	86	2486 (avg. of 19 audios per item)	149,161
Reviews	8	458	23,374

Table 1. Information gathered from RSS feeds is stored into a historic relational database.

Table 1 shows some basic statistics of the data that has been gathered since mid April, 2005 until the first week of July, 2006 (except for the album reviews that started in mid June, 2005). These numbers show that the system has to deal with a daily fresh incoming data.

[14] http://www.sdreader.com/
[15] http://www.subpop.com/
[16] http://www.75orless.com/
[17] http://www.rollingstone.com/

On the other hand, we have defined a music ontology[18] (OWL DL) that describes basic properties of the artists and the music titles, as well as some descriptors extracted from the audio (e.g. tonality —key and mode—, ryhthm —tempo and measure —, intensity, danceability, etc.). In [2] we propose a way to map our ontology and the MusicBrainz ontology, within the MPEG-7 standard, that acts as an upper-ontology for multimedia description.

A focused web crawler has been implemented in order to add instances to the music ontology. The crawler extracts metadata of artists and songs, and the relationships between artists (such as: "related with", "influenced by", "followers of", etc.). The seed sites to start the crawling process are music metadata providers[19], and independent music labels[20]. Thus, the music repository does not consist only of mainstream artists.

Based on the music ontology, the example 1.1 shows the RDF/XML description of an artist from *Garageband.com*.

```
<rdf:Description rdf:about="http://www.garageband.com/artist/
   randycoleman">
 <rdf:type rdf:resource="&music;Artist"/>
 <music:name>Randy Coleman</music:name>
 <music:decade>1990</music:decade>
 <music:decade>2000</music:decade>
 <music:genre>Pop</music:genre>
 <music:city>Los Angeles</music:city>
 <music:nationality>US</music:nationality>
 <geo:Point>
   <geo:lat>34.052</geo:lat>
   <geo:long>-118.243</geo:long>
 </geo:Point>
 <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 1.1. Example of an artist individual

Example 1.2 shows the description of a track individual of the above artist:

```
<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 rdf:resource="http://www.garageband.com/
    artist/randycoleman" />
```

[18] The OWL DL music ontology is available at: http://foafing-the-music.iua.upf.edu/music-ontology#

[19] Such as http://www.mp3.com, http://music.yahoo.com, http://www.rockdetector.com, etc.

[20] E.g. http://www.magnatune.com, http://www.cdbaby.com and http://www.garageband.com

```
 <music:duration>247</music:duration>
 <music:key>D</music:key>
 <music:keyMode>Major</music:keyMode>
 <music:tonalness>0.84</music:tonalness>
 <music:tempo>72</music:tempo>
</rdf:Description>
```

Listing 1.2. Example of a track individual

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

3.2 Music Recomendation process

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

1. Get music related information from user's FOAF interests, and user's listening habits
2. Detect artists and bands
3. Compute similar artists, and
4. Rate results by relevance.

In order to gather music related information from a FOAF profile, the system extracts the information from the FOAF interest property (if dc:title is given then it gets the text, otherwise it gathers the text from the title tag of the resource).

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 artists' individuals 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.

The system offers two ways of recommending music information. *Static* recommendations are based on the favourite artists encountered in the FOAF profile. We assume that a FOAF profile would be barely updated or modified. On the other hand, *dynamic* recommendations are based on user's listening habits, which is updated much more often that the user's profile. With this approach the user can discover a wide range of new music and artists.

Once the recommended artists have been computed, *Foafing the Music* filters music related information coming from the gathered information (see section 3.1) in order to:

- Get new music releases from iTunes, Amazon, Yahoo Shopping, etc.
- Download (or stream) audio from MP3–blogs and Podcast sessions,
- Create, automatically, XSPF[21] playlists based on audio similarity,

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

- Read Artists' related news, via the PubSub.com server[22]
- 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 the above mentioned functionalities, there is a feed subscription option to get the results in the RSS format.

4 Conclusions

We have proposed a system that filters music related information, based on a given user's profile and user's listening habits. A system based on FOAF profiles and user's listening habits allows to "understand" a user in two complementary ways; psychological factors —personality, demographic preferences, socio-economics, situation, social relationships— and explicit musical preferences. In the music field context, we expect that filtering information about new music releases, artists' interviews, album reviews, etc. can improve a recommendation system in a dynamic way.

Foafing the Music is accessible through http://foafing-the-music.iua.upf.edu

Acknowledgements

This work is partially funded by the SIMAC IST-FP6-507142, and the SALERO IST-FP6-027122 European projects.

References

1. O. Celma, P. Cano, and P. Herrera. Search sounds: An audio crawler focused on weblogs. In *Proceedings of 7th International Conference on Music Information Retrieval*, Victoria, Canada, 2006.
2. R. Garcia and O. Celma. Semantic integration and retrieval of multimedia metadata. In *Proceedings of 4rd International Semantic Web Conference. Knowledge Markup and Semantic Annotation Workshop*, Galway, Ireland, 2005.
3. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.
4. G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 4(1), 2003.
5. F. Pachet. *Knowledge Management and Musical Metadata.* Idea Group, 2005.
6. 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.
7. A. Uitdenbogerd and R. van Schnydel. A review of factors affecting music recommender success. In *Proceedings of 3rd International Conference on Music Information Retrieval*, Paris, France, 2002.

[22] http://www.pubsub.com