Getting music recommendations and filtering newsfeeds from FOAF descriptions

Oscar Celma[1], Miquel Ramírez[1], and Perfecto Herrera[1]

Music Technology Group, Universitat Pompeu Fabra, Barelona, SPAIN
http://www.iua.upf.es/mtg

Abstract. This document proposes to use the *Friend of a Friend* (FOAF) definition to recommend music depending on user's musical tastes and to filter music-related newsfeeds. One of the goals of the project is to explore music content discovery, based on both user profiling —FOAF descriptions— and content-based descriptions —extracted from the audio itself.

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, straight-forward acquisition, the value of these resources suffers from a lack 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 pieces. Music is an important vehicle for telling 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 between them and the things they create and do. The FOAF (*Friend Of A Friend*) project[1] provides conventions and a language "to tell" a machine the sort of things a user says about herself in her homepage. FOAF is based on the RDF/XML[2] vocabulary. We can foresee that using user's FOAF profile would allow a system to better understand user musical needs.

The main goal of SIMAC[3] project is doing research on semantic descriptors of music contents, in order to use them, by means of a set of prototypes, for providing song collection exploration, retrieval and recommendation services. These services are meant for "home" users, music content producers and distributors and academic users. One special feature is that these descriptions are composed by *semantic descriptors*. Music will be tagged using a language close to the user's own way of describing its contents —moving the focus from low-level to higher-level (i.e. semantic) descriptions.

[1] http://www.foaf-project.org
[2] http://www.w3.org/RDF
[3] http://www.semanticaudio.org

2 Background

Recommender Systems are software applications whose purpose is to deliver information to people that "needs" it. Put this way, one cannot tell the difference between a Recommender System and a Search Engine —both software types share the same purpose: to select objects (or items) from a repository whose features were found to satisfy the querying user's needs.

However, there exist two subtle but meaningful differences between "Search Engines" and "Recommender Systems". The first of these differences lies in the design intention, or better said: the wording of the problem to address when designing the system. Is that "information need" related to solving a contingent situation, or is that need something periodic or steady? The second one is also another design intention difference which lies in the use of two different words to describe the system: does it *retrieve* information from a relatively static repository of information? Or does it *filter* objects embedded in an incoming stream of information?

The "Recommender System" term emerged as a logical evolution of the research of information retrieval (IR) systems. This evolutive main feature was the emphasis put on the "query" concept definition and representation. Recommender Systems were initially thought as *information filtering* systems, whose technological framework baseline stemmed from *information retrieval* systems [1]. This then, effectively implies that a Recommender System is an inherently dual purpose application: the user profiling of steady information needs might be used to better understand and attend immediate, unforeseen needs.

There are two main approaches to recommend items to users: collaborative filtering and content-based filtering. Next section explains the differences between both approximations.

2.1 Collaborative filtering versus Content-based filtering

Collaborative filtering consists of making use of feedback from users to improve the quality of 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. One of the main caveats of this approach is the fact that the only way to recommend brand new items is that some user has to previously rate or review that item. are some examples that succeed based on this approach. For instance, Amazon is a good illustration system [2].

On the other hand, content-based filtering tries to extract useful information —from the items of the user's collection— that could be useful to represent user's needs. This approach solves the limitation of collaborative filtering as it can recommend new items (even before not knowing anything from that item), by comparing the actual set of user's items and calculating the distance with some sort of similariy measure. In the music field, to extract musical semantics from the raw audio and computing similarities between music pieces is a challenging one. Traditional music similarity measures use low-level —mainly

timbre-based— features. We belive that adding cultural metadata terms to such a similarity measure can help to get better results.

2.2 Music recommendation systems

The main goal of a music recommendation system is to propose interesting and unknown music artists (and their available tracks —if possible—) to the end-user, based on her musical taste. But musical taste and music preferences are affected by several factors, even demographic and personality traits. Then, combining music preferences and personal aspects —such as: age, gender, origin, occupation, musical education, etc.— could improve music recommendations [3].

Moreover, a music recommendation system should be able to get new music dynamically, as it should recommend new items to the user once in a while. In this sense, there is a lot of free 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. iTunes Music Store[4] offers the possibility to subscribe to its *New Music Tuesdays* system, via email. This service issues one email message every week with exclusives, live session recordings, remixes, celebrity playlists, and unreleased tracks from their artists. iTunes provides, as well, an RSS (version 2.0) feed generator[5], with an hourly updated period, that publishes new releases as they are made available. A music recommendation system should take advantage of these publishing services, as well as integrate it into the system, to filter and recommend new music to the user.

Most of the current music recommenders are based on collaborative filtering approach, or an hybrid version including clustering and users' communities Examples of such systems are: Audioscrobbler[6], iRate[7], Goombah Emergent Music[8] and inDiscover[9]. The basic idea of a music recommender system based on collaborative filtering is to keep track of which artists a user listens —through WinAmp or XMMS plugins—, to in order to finding other users with similar tastes and, finally, recommending similar artists to the user, according on these similar listeners' taste. But, digital music collections can be huge (thousands of files), and very heterogeneous. Thus, this approach to recommend music can generate some "silly" (or obvious) answers.

The main goal of our prototype system is to recommend, to discover and to explore music content; based on both user profiling —via FOAF descriptions— and content-based descriptions —extracted from the audio itself.

[4] http://www.apple.com/itunes
[5] http://phobos.apple.com/WebObjects/MZSearch.woa/wo/0.1
[6] http://www.audioscrobbler.com
[7] http://irate.sourceforge.net
[8] http://goombah.emergentmusic.com/
[9] http://www.indiscover.net/

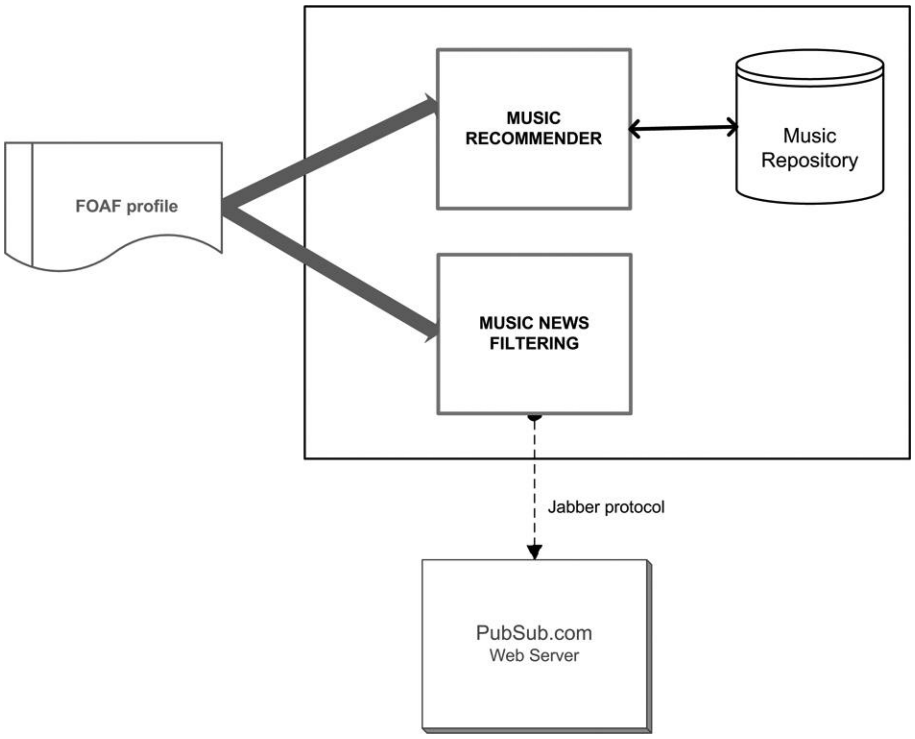

Fig. 1. System overview.

The system is composed by two main components. The first component is the music recommender, while the second is the (music related) newsfeeds filtering. Both components are based on user's FOAF profile (example 3.1 shows a possible input file[10]). Example 3.1 Next sections explains each component of the system.

3 System overview

3.1 Music recommender

Music recommendations are done through the following steps:

1. Get interests from user's FOAF profile
2. Detect artists and bands
3. Access to Music repository and select related artists, from artists encountered in the user's FOAF profile
4. Rate results by relevance

[10] A real example extracted from *http://www.livejournal.com*, only changing user's name

```
<rdf:RDF xml:lang="en">
  <foaf:Person>
    <foaf:nick>test_user</foaf:nick>
    <foaf:dateOfBirth>04-17</foaf:dateOfBirth>
    <foaf:mbox_sha1sum>
      ce24ca1400c2f511c652b015a1f064dda8356f9a
    </foaf:mbox_sha1sum>
    <foaf:page>
      <foaf:Document
        rdf:about="http://www.livejournal.com/userinfo.bml?user=test_user">
        <dc:title>LiveJournal.com Profile</dc:title>
      </foaf:Document>
    </foaf:page>
    <foaf:weblog rdf:resource="http://www.livejournal.com/users/test_user/"/>
    <foaf:interest dc:title="gretsch"
      rdf:resource="http://www.livejournal.com/interests.bml?int=gretsch"/>
    <foaf:interest dc:title="pub"
      rdf:resource="http://www.livejournal.com/interests.bml?int=pub"/>
    <foaf:interest dc:title="dogs d'amour"
      rdf:resource="http://www.livejournal.com/interests.bml?int=dogs+d%27amour"/>
    <foaf:interest dc:title="social distortion"
      rdf:resource="http://www.livejournal.com/interests.bml?int=social+distortion"/>
    <foaf:interest dc:title="beer"
      rdf:resource="http://www.livejournal.com/interests.bml?int=beer"/>
    <foaf:interest dc:title="the misfits"
      rdf:resource="http://www.livejournal.com/interests.bml?int=the+misfits"/>
    <foaf:interest dc:title="the pogues"
      rdf:resource="http://www.livejournal.com/interests.bml?int=the+pogues"/>
    <foaf:interest dc:title="whiskey"
      rdf:resource="http://www.livejournal.com/interests.bml?int=whiskey"/>
  </foaf:Person>
</rdf:RDF>
```

Example 3.1: Example of a user's FOAF profile

The prototype reads an input FOAF profile —that is, an RDF file—, and extracts user's interests. Then queries to a music repository to detect whether the interest is a music artist (or a band) and selects similar artists to the ones found. To get artists' similarities, a focused web crawled has been implemented to look for relationships between artists (such as: related with, influenced by, followers of, etc.). This web crawler has gathered information from several music portals, such as: *allmusic.com*, *mp3.com* and *msn.music.com*, as well as some sites that contains information (and audio) from "unknown" artists: *magnatune.org*, *garageband.com* and *vitaminic.com*. All these information has been stored into our music repository.

Moreover, a music similarity distance is used to recommend tracks that are similar to tracks composed or played by artists found in the FOAF profile. Tracks

are filtered to the user depending on automatically extracted music content descriptions. These descriptions are composed by a certain number of quantificable measures taken directly from the track samples ([4], [5]). Currently, we are considering *(i)* ryhthm descriptors: tempo (beats per minute), meter (binary or ternary) and danceability, *(ii)* tonal descriptors: tonality, mode and tonality strength, and *(iii)* timbre descriptors: global loudness and several low-level timbre descriptors. These audio descriptions are evaluated according to a preference model derived from the analysis of users' listening patterns. Euclidean distance —from the instances of the user's model and the potential recommended tracks— is computed to determine whether an incoming track is likely to be listened by the user.

Based on the FOAF example (see example 3.1), the prototype detects the following artists from the user's profile: *Dogs d'Amour*, *Social Distortion*, *The Misfits* and *The Pogues*. Starting from these artists, the system searches for similar artists and artists influenced by them, and scores them in terms of counting artist occurrences. If there are any tracks in the music repository from artists in the FOAF profile, it computes the similarity and gets the most significant similar tracks from other artists. Figure 2 shows the ouput recommended artists.

To our knowledge, nowadays it does not exist any system that recommends items to a user, based on her FOAF profile. There is the FilmTrust system[11] which is a part of a research study to understand how social preferences might help web sites present information to users in a more useful way. The system collects user reviews and ratings about movies, and holds it into the user's FOAF profile. Although it has not yet implemented a recommendation system, it includes a rating algorithm for films based on a trust-based algorithm [6].

3.2 Music related news filtering

Filtering news based on user's profile is another issue related with recommender systems. This kind of system is designed to filter mail, messages from mailing lists, Internet News articles, newswire stories, etc.

In our system, the music related news filtering component queries a newsfeeds system that filters news regarding to related artists found in user's FOAF profile. To do so, this component permits to communicate with the PubSub server[12], via the Jabber protocol, and creates an RSS feed with a given query —that is the user musical preferences found in the FOAF file. PubSub is a matching service that instantly notifies a user when new content is created that matches user's subscription. PubSub reads over over 8 million weblogs, more than 50,000 internet newsgroups and all SEC (EDGAR) filings. Jabber[13] is an open secure protocol, an ad-free alternative to consumer instant messaging services like ICQ, MSN, and Yahoo. Jabber makes use of XML protocols that enable any two entities on the Internet to exchange messages, presence, and other structured information in (close to) real time.

[11] http://trust.mindswap.org/FilmTrust
[12] http://www.pubsub.com
[13] http://www.jabber.org

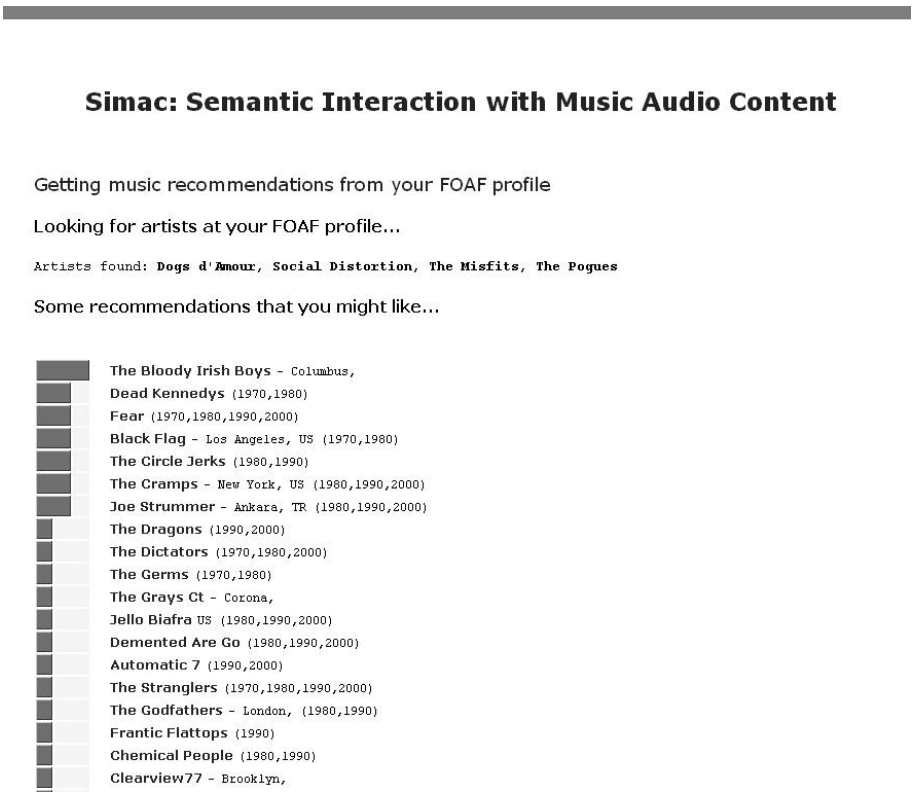

Fig. 2. Recommended artists from artists detected in a user's FOAF profile.

Once the subscription with PubSub.com has been created, it is possible to visualize all the music related news for a given user. Each news item has a bar score that shows how much it is related with user's musical interests. Scoring is done using the TF/IDF ranking algorithm [7]. TF/IDF ranks documents by counting the number of ocurrences of user's term query into each document.

4 Implementation

The prototype is completely written in PHP. It uses several PHP modules to implement each subsystem:

– **FOAF**: To process RDF files it uses the FOAF class from the Pear framework (PHP Extension and Application Repository)[14]. This FOAF class makes

[14] http://pear.php.net/

use of the RAP module[15], thus is easy to access to the document using the RDQL language.
- **RSS**: Newsfeed processing is done using MagpieRSS[16] module. MagpieRSS is an RSS and Atom parser for PHP.
- **Jabber protocol**: To interact with PubSub.com server a PHP class has been implemented. This class allows to connect to the server, to authenticate, and to create, retrieve and delete a subscription. This class makes use of *class.jabber.php*[17] module developed by Nathan Fritz.

Access to the prototype is available at: http://www.semanticaudio.org/foafin-the-music

5 Conclusions

We have proposed a system that recommends music and filters music related news based on a given user's profile. A system based on FOAF profiles allows to "understand" a user in two complementary ways; psychological factors — personality, demographic, socio-economics, situation— and explicit musical preferences. This system, then, is able to filter and to contextualize users' queries.

In the music field context, we expect that using news filtering about music new releases, artists' interviews, album reviews, etc. can improve a recommendation system in a dynamic way. Finally, this approach opens a wide range of possible usages and applications, such as notifying a user the forthcoming gigs by an artist —playing close to user's location— whose music is similar to user's musical taste.

6 Acknowledgments

This work is partially funded by SIMAC IST-FP6-507142 European project.

References

1. Nicholas J. Belkin and W. Bruce Croft: Information Filtering and Information Retrieval: Two sides of the same coin?, Communications of the ACM. December, 1997, Volume 35, number 12, pages 29-39.
2. Greg Linden and Brent Smith and Jeremy York: Amazon.com Recommendations: Item-to-Item Collaborative Filtering, IEEE Internet Computing, Volume 4, number 1,2003.
3. Alexandra Uitdenbogerd and Ron van Schyndel: A Review of Factors Affecting Music Recommender Success, ISMIR 3rd International Conference on Music Information Retrieval, October 13-17, 2002.

[15] http://www.wiwiss.fu-berlin.de/suhl/bizer/rdfapi
[16] http://magpierss.sourceforge.net/
[17] http://cjphp.netflint.net/

4. Gouyon, F. Dixon, S. A Review of Automatic Rhythm Description Systems, Computer Music Journal, Vol. 29, Issue 1 - Spring 2005
5. Gomez, E. Herrera, P. Estimating The Tonality Of Polyphonic Audio Files: Cognitive Versus Machine Learning Modelling Strategies, Proceedings of the 5th International ISMIR 2004 Conference, October 2004. Barcelona, Spain.
6. Golbeck, Jennifer, Bijan Parsia: "Trust Network-Based Filtering of Aggregated Claims", to appear in the International Journal of Metadata, Semantics, and Ontologies, 2005.
7. Ricardo Baeza-Yates and Berthier Ribeiro-Neto,: ACM Press, Modern Information Retrieval, Addison-Wesley, 1999.