A Prototype for Visualizing Music Artist Networks

Bruno G. Costa[(1)], Fabien Gouyon[(2)] and Luís Sarmento[(3)]

[(1)] Universidade Católica Portuguesa, Porto, Portugal

[(2)] INESC Porto, Portugal

[(3)] LIACC / FEUP, Universidade do Porto, Portugal

Abstract* — This paper reports on a prototype providing a simple yet efficient interface to navigate through networks of music artists. Built upon data gathered from Last.fm, it provides two simultaneous layers of information: (i) a graph built from artist similarity data, and (ii) overlaid labels containing user-defined tags. Differing from existing artist network visualization tools, the proposed prototype emphasizes commonalities as well as main differences between artist categorizations, hence providing richer browsing to the user. The prototype can be accessed at *http://pattie.fe.up.pt/rama

***Index Terms* — Music, Visualization, User interfaces**

I. Introduction

With the advent of worldwide computer networks, the explosion of computer storage capacity, the dramatic increase of information flows and computing power, together with the improvement of computer graphics, all fields of science face the challenge of dealing with and visualizing massive amounts of data. Consequently, in the last decade, a great deal of work has been dedicated to interface design for visual representation of data, hence opening new avenues for *data exploration* (navigation, browsing, discoveries), *computation* (analysis of data), *communication* (summarizing what otherwise would need many words to be understood) or *aesthetic purposes* [2]. As other research fields, Sound and Music Computing [1] has witnessed a recent interest in the visualization of music data, notably within the realm of Music Information Retrieval applications (as e.g. personalized music recommendation, music database browsing, online music access, query-based music retrieval, automatic play-list generation, etc.). Recent efforts include the visualization of the structure of music pieces,[1] flows of music data in networks of music generating elements on top of tangible multi-touch interfaces [3], or browsing interfaces for artist recommendation [4]. A popular metaphor for information visualization is that of networks, or connected graphs, where the data is presented through nodes and lines connecting them which represent a relationship between items. Nodes can stand for music genres,[2] albums[3] or songs.[4] And, more related to this paper, nodes can also represent artists while connecting lines represent artist similarity. This is the case of a number of prototypes, as e.g. LivePlasma[5] or TuneGlue,[6] using artist similarity data (i.e. recommendations) from Last.fm or Amazon. Similarly to these two applications, our prototype uses third-party similarity data to display music artists in 2D connected networks. This has the clear advantage over simple list-based recommendations –as those provided by Last.fm or Amazon– to visualize artist connections at a distance higher than 1 (further away than direct links, one can embrace in the same visualization artists that are similar to the query as well as those similar to the answers, and so on). In our prototype, we sought a good balance between readability (avoiding cluttered use of space) and richness of the data presented to the user. Hence the special focus, in the design phase, on a proper use of graphical features (e.g. colors and transparencies) as well as interactivity between the user and the prototype (some information is shown by default, some other only as results of the user's interactions). An originality of our prototype is to also place a second layer of information, allowing the simultaneous visualization of artists and corresponding user-defined tags. This allows emphasizing both the common or distinctive user opinions about an artist and its neighbors.

II. Last.fm

Last.fm (http://www.last.fm) is an internet-based social music platform, where users can listen to music, find information about artists they like, or discover artists they might not know. Following the Web 2.0 concept, users can also set up their own profile, facilitating targeted automatic recommendations, among other things they can

[1] http://www.turbulence.org/Works/song/

[2] http://techno.org/electronic-music-guide

[3] http://www.dimvision.com/musicmap

[4] http://musicovery.com/

[5] http://www.liveplasma.com

[6] http://audiomap.tuneglue.net

also get information about users with similar tastes, gigs in their local area, videos, etc. Last-fm site provides extensive encyclopedic information about artists. Such information (see Figure 1 for an example) is collaboratively edited by the user.

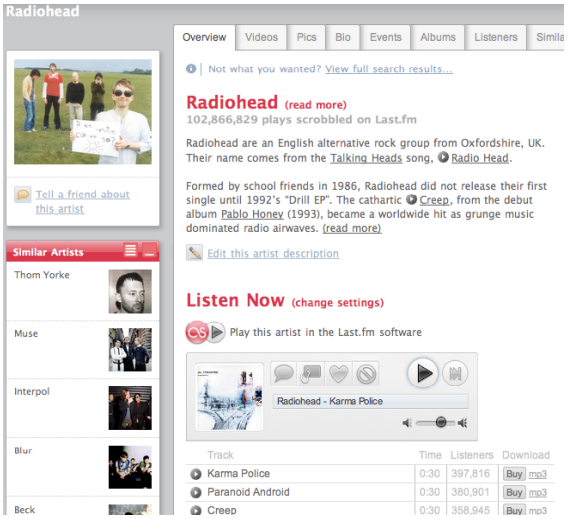

Figure 1. Last.fm entry webpage for the band "Radiohead" (http://www.last.fm/music/Radiohead)

User profiles –and hence recommendations– are constantly updated via a software (free of use) which gathers ("scrobbles" in the Last.fm vernacular) statistics about the music listened to by users. User listening patterns are recorded and analyzed by Last.fm in order to better organize and recommend music.

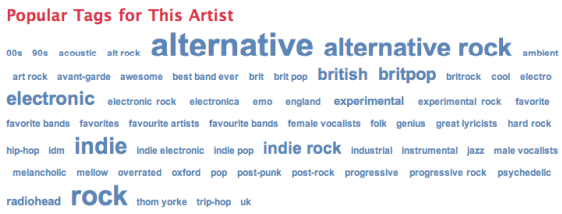

Figure 2. Full list of user tags for the band "Radiohead"

Users are also encouraged to organize the music they listen to by assigning "tags" (i.e. free-text keywords, or labels) to artists, or even specific albums or tracks. The definition of tags is up to the users and can describe any aspect users believe are relevant, as music genres (e.g. "rock", "Viking metal"), locations (e.g. "Berlin"), mood (e.g. "chill"), opinions (e.g. "songs my mother would like"), contexts (e.g. "love") or just about anything that cross users' minds (see and Figure 2 for examples of tags assigned to the band "Radiohead"). Tagging music helps users to browse their music. But the real power of tags becomes clear when considering that tags of hundreds of thousands of users are combined, providing an emerging "bottom-up" categorization of music.

A cornerstone of Last.fm functionalities resides in links of similarity between artists (which can be seen on the left column of Figure 1 and on Figure 3, and which is central to automatic recommendations made to users). The algorithm used for computing similarities between artists is unknown (to the authors of this paper) but is probably based on (i) the analysis of user-added tags, on (ii) user listening patterns such as co-occurrence statistics (users that listen to artist X also listen artist Y), and on (iii) user profiles information ("similar " users should like "similar" artists).

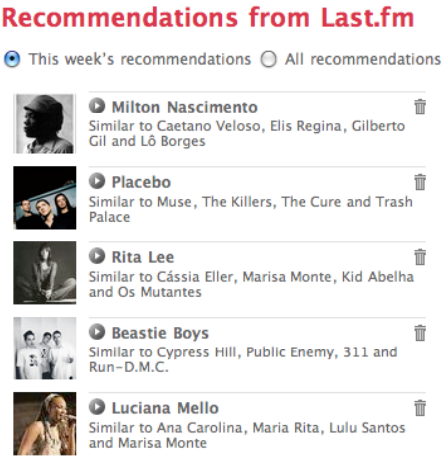

Figure 3. Artist similarities and personal artist recommendations made by Last.fm to one of the authors of this paper

III. System Overview

Our system uses data available through Last.fm web services API.[7] Available data categories are "User Profile data", "Artist Data", "Album Data", "Track Data", "Tag Data", "Group Data", "Forum Data" and "Geo-aware Data" (more details can be found on the referred site). From all the data available, we mainly focus on data specially concerning a given artist such as (i) the list of the most similar artists, (ii) tags that users have provided for such artist, and (iii) a value indicating the artist "popularity" within Last.fm community. Some of the data obtained from Last.fm web services is already normalized. For example, each similar artist to a given artist is assigned a weight ranging from 100 (full similarity) to 1 (almost no similarity). Each user-defined tag is also given a weight to quantify the association level: 100 means full association, while 1 indicates loose association.

[7] See http://www.audioscrobbler.net/data/webservices/

Because each individual access the Last.fm web service involves a considerable overhead due to network latency and server load, we created a local copy of the data we needed for a total number of 583.000 artists. This improves the global performance of our system since most accesses are made locally, but requires an extra pre-processing step.

Our system is built on top of a client/server architecture (Figure 4 provides a general overview). The visualization is performed on the client side (the user application) using information obtained from the server via an HTTP request. The server side manages the data that has been crawled in the pre-processing stage and performs all the necessary operations to provide the client with the data needed for the visualization.

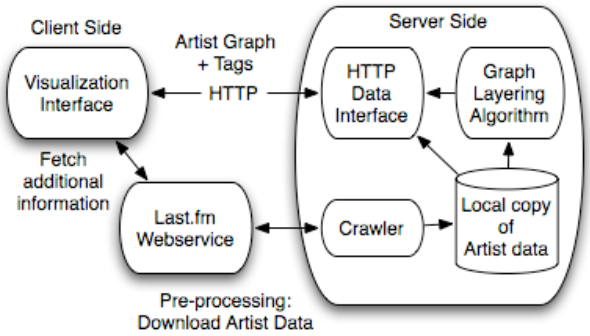

Figure 4 - System overview

Given a specific user query (i.e. an artist name) submitted by the user through the visualization interface), the server provides a list containing information that will allow the visualization application to draw the related artist network. For *each* artist in the network (the original queried artist and the top similar artists found, the information given is (i) the 2D coordinates of the corresponding node in the graph, (ii) the list of user-defined tags, and (iii) explicit similarity relationships with other artists in the network (for drawing the necessary edges). Node positions in the graph are computed in the server-side using a force-directed strategy. Artist nodes are considered to be connected by springs whose elastic constant is proportional to their pair-wise similarity (as given by the data fetched from Last.fm site during pre-processing). A new graph layout is computed in real time for each query, even if the query has been processed before. All the information is sent to the visualization interface in text format to allow a simple parsing procedure.

Figure 5 shows a snippet of a possible answer for the query "Radiohead". 2D coordinates and similar artists of, respectively, "Radiohead" (the original artist) and "Sigur Rós" (one of the similar artists found) are in bold. User tags are ordered by relevance for these specific artists (e.g. "alternative" is more relevant than "electronica" for "Radiohead").

```
[…]
Radiohead::http://userserve-ak.last.fm/serve/160/2156872.jpg::1325099
alternative::100::rock::83::alternative rock::59::indie::55::electronic::38
::britpop::31::british::27::indie rock::27::experimental::15::seen live::13
::Progressive rock::8::electronica::8::90s::6::post-rock::5::pop::4
::Experimental Rock::3::UK::3::art rock::3::psychedelic::3::00s::2
48.96::44.77::Coldplay::Sigur Rós::Muse::Beck::Thom Yorke
[…]
Sigur Rós::http://userserve-ak.last.fm/serve/160/96448.jpg::282210
post-rock::100::ambient::73::icelandic::51::indie::44::alternative::41::electronic::26
::post rock::19::seen live::15::experimental::14::rock::9::shoegaze::8::chillout::6
::indie rock::5::electronica::5::alternative rock::4::Progressive rock::4::instrumental::4
::ethereal::2::atmospheric::2::psychedelic::2
53.67::68.31::Múm::Mogwai
[…]
```

Figure 5. Example server answer to a query

The visualization interface also uses additional data fetched from the Last.fm website at runtime, namely artist pictures. Unlike tag, similarity and popularity data, here performance constraints are not severe (an extra access for each artist).

IV. User Interface

The user interface, done in Processing,[8] queries our HTTP data interface and retrieves artist graph coordinates and tag information. Basically, a textbox enables users to enter the name of an artist –that is, the query–, and names of a set of artists are consequently drawn on a map at their respective 2D coordinates, links between similar artists are materialized as leaves. The size of the leaves corresponds to artist popularity.

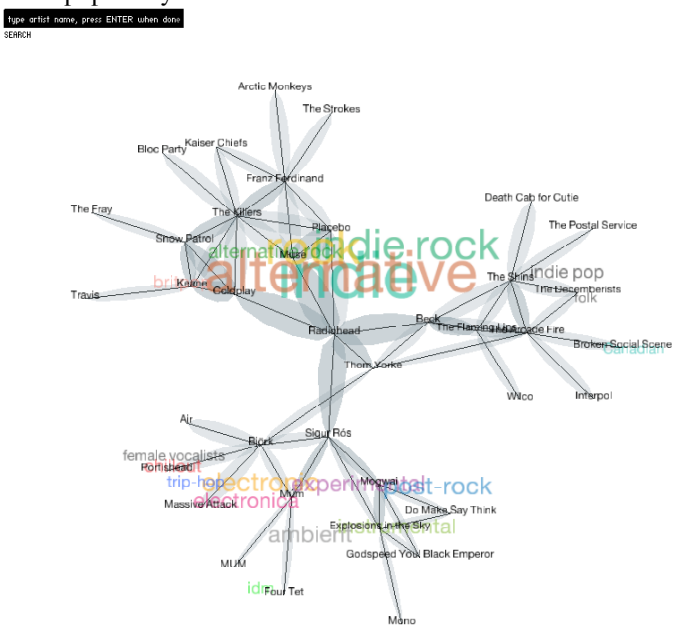

Figure 6. User interface (query: "Radiohead").

[8] http://processing.org/

Moving the mouse cursor on top of a specific artist results in the presentation of diverse data: the artist picture (gathered at run-time from the Last.fm site) and a link to its Last.fm webpage.[9] Left-clicking on any artist name has the same effect as entering its name in the query box, i.e. sending a new query and refreshing the map with this query as seed. This allows simple user navigation through the artist network.

Tags that are common to a number of artists in the map are represented on top of them. Tag sizes are proportional to the number of artists for which they are relevant. This information is shown by default, no action being required from the user. This allows the identification –and further exploration– of regions of the network where artists have been predominantly characterized by specific tags. For instance, in the network of "Radiohead" similar artists (on Figure 6), one can identify a region of artists tagged as "alternative" in the center, while the region on the right branch is predominantly tagged "indie pop."

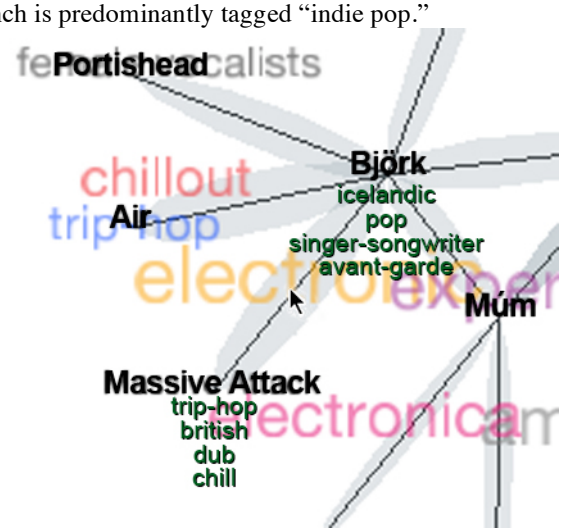

Figure 7. Example of *specific tags*: The artists "Björk" and "Massive Attack" are both similar to some degree to "Radiohead" (initial query). They are both in a region characterized by "electronic", however, "icelandic", "pop", "singer-songwriter" and "avant-garde" apply to the former and not to the latter, while "trip-hop", "brittish", "dub" and "chill" apply to the latter and not the former.

An additional original feature of our user interface resides in the possibility to visualize tags that are *specific* to an artist, i.e. that are relevant to it but *not* relevant to its neighbors. This information is shown only as the result of an interaction between the user and the application. Crossing with the mouse cursor the link between 2 connected artists results in the rendering of 2 sets of tags: those that are specific to each of these 2 artists, and therefore emphasizes differences between them (see Figure 7).

[9] Indeed, the goal here is not to duplicate a Last.fm artist page, but rather to provide a complementary view of some of its content.

IV. Conclusions and Future Work

The current prototype provides a simple yet efficient interface to navigate through networks of similar artists, allowing users to obtain a richer view of artists they know, and to easily discover new bands and artists that they might like. It provides two simultaneous layers of information: (i) a graph built from artists and their connections, and (ii) overlaid labels containing user-defined tags that express the classification made by the Last.fm community for each of the artists. From experimentation we have observed that the system effectively allows to identify clusters of tightly connected bands and artists (such as for example former members of a band that pursued a solo career). Additionally, our visualization procedure also emphasizes the main differences between artists, allowing the user also to check what are the most distinctive attributes of artists within the similarity network.

Future work includes enhancing user experience by adding song snippets for each artist, so that the user can play them on demand while navigating across the network. Also, we plan to improve interactivity by allowing the user to optionally navigate through user-defined tags, and not just artists. We will also focus on allowing the user to manipulate the graph (zooming, rotating, etc) and to edit it (adding and removing nodes (artists) from the graph).

Acknowledgement

This work was partially supported by grant SFRH/BD/23590/2005 from FCT (Portugal), co-financed by POSI.

References

[1]. N. Bernardini, X. Serra, M. Leman, G. Widmer, G. De Poli, (eds) "A Roadmap for Sound and Music Computing", 2007 (http://smcnetwork.org/roadmap)

[2]. M. Hearst. "User interfaces and visualization." In R. Baeza-Yates and B. Ribeiro-Neto, editors, *Modern information retrieval*. Harlow, Essex: ACM Press, 1999

[3]. S. Jordà,, M. Kaltenbrunner, G. Geiger and R. Bencina, R. "The Reactable*", *Proceedings of the International Computer Music Conference,* 2005

[4]. E. Pampalk and M. Goto "MusicSun: A New Approach to Artist Recommendation*." Proceedings of the International Conference on Music Information Retrieval*, 2007