MUSIC DISCOVERY IN SPOTIFY WITH RAMA

José Bateira[1,2], **Fabien Gouyon**[2], **Matthew Davies**[2], **Marcelo Caetano**[2]
[1]DEI-FEUP
[2]Sound and Music Computing Group, INESC TEC
ei10133@fe.up.pt {fgouyon,mdavies,mcaetano}@inesctec.pt

ABSTRACT

This demo presents the Spotify App [1] of RAMA [2] that aims to improve the way music recommendations are shown in Spotify [3]. This version implements most of the current features of RAMA's original website [4]: edition of the visualization parameters (to obtain more detailed graphs), edition of the graph by adding and removing node artists (for users that want to meticulously edit the graph) and visualization of the tags/genres that describe an artist in the graph. The metadata used throughout the application is provided by Spotify's and Echonest's API [5].

1. MUSIC DISCOVERY TOOLS IN SPOTIFY

Spotify provides different features for the users to discover new music such as Browse, Activity, Discovery, Radio and Spotify Apps. These features commonly present the recommendations as a list and/or grid, which is not ideal because the user cannot understand the reason behind such recommendations. Spotify Apps gives different possibilities for the user to discover new music because they rely on third-party applications. For example, Last.fm [6] developed their application that uses its recommendation system and this way Spotify users can have the best of both worlds from Spotify and Last.fm all in one application. But the way the recommendations are displayed (lists and grids) remains an issue. This demo showcases a system that integrates RAMA into Spotify, providing a visual representation that is more natural and intuitive than lists.

2. RAMA IN SPOTIFY

RAMA uses a tree graph to display a network of music artists that share similarities. There are other visual-based recommendation applications (for example, `http://liveplasma.com` and `http://musicroamer.com`). A downside of the applications mentioned above is the use of Youtube [7] to stream the music tracks. This is not ideal given that the sound quality is not appropriate for music streaming. Also Youtube is not a music oriented service thus music-related queries do not always return the expected content. For example, interviews or podcasts might appear in the search results among the music videos.

In contrast, Spotify is a music oriented service that has a large music catalogue and a high quality music stream. It also has a very user-friendly environment that is ideal for the integration with RAMA. Spotify has made several tools available to develop applications [8] integrating their services. Spotify Apps is the ideal choice to integrate RAMA into Spotify thus improving user experience in a more natural and intuitive way.

3. MAIN FEATURES

3.1 Visualization of the Artists' Map

RAMA's Spotify App automatically draws the map with the current playing artist as the main node, as seen in Figure 1. The graph-like structure is created by recursively

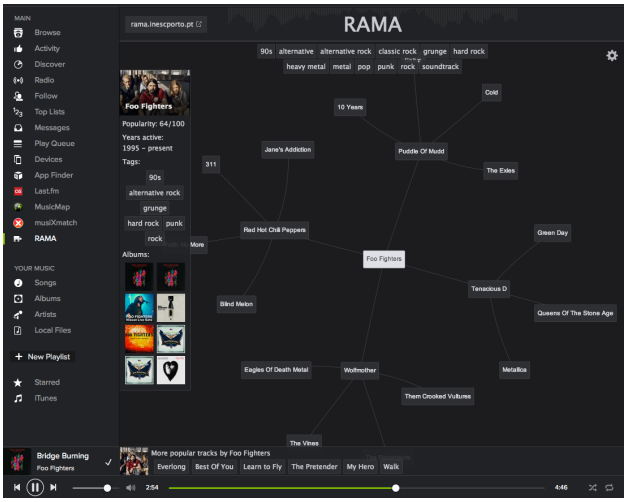

Figure 1. RAMA Spotify App opened inside the Spotify Desktop Client

[1] `https://developer.spotify.com/technologies/apps`
[2] `http://rama.inescporto.pt`
[3] `http://spotify.com`
[4] `http://rama.inescporto.pt/app`
[5] `http://developer.echonest.com`
[6] `http://last.fm`
[7] `http://youtube.com`
[8] `http://developer.spotify.com`

fetching a list of related artists from each artist using Spotify's API [9] .

3.2 Visualization Parameters

The visualization parameters can be personalized by the user in the settings menu in order to create richer graphs that match the user's preferences better. The depth value of a graph determines how deep the recursive algorithm goes when constructing the graph, meaning, depth is the maximum distance between the root node and any other node in the graph. The branching value of a tree graph determines the maximum number of child nodes a node can have. The tree mode option indicates whether the graph to be built is a tree graph or not. This means that when the tree mode is on, the graph has less edges (which leaves the graph more clean) and so it becomes a tree graph.

3.3 Graph Edition

The available features to edit the graph are: expand node, delete node and create a new map. These interactions are available in the Artist Menu (Figure 2). The expand node action allows the user to expand a node further (ignoring the graph's branching value). The delete node action allows the user to delete a node from the graph. These two actions are useful for when the user wants to construct the graph to his/her needs. The new map action lets the user create a whole new graph from a another node. This way the root node will be the selected node of the new graph.

3.4 Artist Menu

The user is able to see additional information about artists in the Artist Menu (Figure 2) such as its popularity value, albums and tags, as well as perform the expand and new map functions described in 3.3. When the user selects a node by clicking on it, the artist menu updates the displayed information. The popularity, the albums and the tags are metadata information retrieved from Spotify's API, although Echonest's API is also used as a fallback source for the tags (sometimes Spotify's API responds with a small number of results).

3.5 Tags Overlay

The tags overlay menu (Figure 3) is meant to enhance the user's understanding on the displayed artists' nodes regarding their tags. These tags are the same ones used in the Artist Menu (Figure 2). They are selectable, and so, when clicked, the respective artist nodes that are described by those tags, are highlighted (as seen in Figure 3). The tags shown in this menu, are just a small sample of the artists' tags of the whole graph.

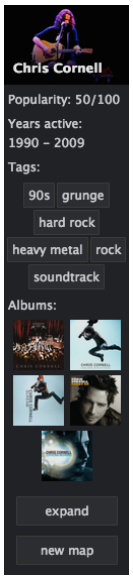

Figure 2. Artist Menu with information about "Chris Cornell"

Figure 3. Tags Overlay

[9] https://developer.spotify.com/docs/apps/api/1.0/api-models-artist.html