

# Integrating multiple distributed data sources to achieve more relevant music search results

Damion Mitchell  
University of Illinois

Huong Luu  
University of Illinois

Joana Trindade  
University of Illinois

## 1. Introduction

According to [1], “music” is the 4th most searched keyword in the internet. In addition, there were approximately 4,558,206 searches for “music” per day over the last month. By looking at these numbers, it is clear that there is need for a component of iWisdm that will properly handle music related queries.

Nevertheless, existing techniques for enhanced search, such as entity annotation, do not apply directly to the music domain. One reason is that it is difficult to devise regular expressions and annotations that describe what music elements, such as a song title and artist or group name, should look like [2]. Furthermore, these elements can be named anything and can be very ambiguous. Examples range from numbers (*e.g.*, the hit “1979” from “Smashing Pumpkins”) to verbs (*e.g.*, “Smile” by “Lily Allen”), and more ambiguous titles, such as “Music”, from “Madonna”.

In this project, we propose the integration of data to produce efficient music queries (*i.e.*, album, artist, and songs) using resources from various websites. Our approach is based on the premise that we can enhance the quality of search results by using a number of available music resources to respond specifically to a user query. In the next sections, we describe our system in terms of input/outputs, functionality, and architecture.

## 2. Proposed System

### 2.1 Description, Input and Output

A component of iWisdm aptly named *iMusic*, which will handle “music” queries for a combination of search keywords, and return a comprehensive set of information applicable to the query in question. These queries can range from searching for an *artist/group* (“Alanis Morissette”), the *title of an album* (“Jagged Little Pill”), or *the title of a song* (“You Oughta Know”). Table 1 below summarizes some of the inputs to be handled by iMusic, and the proposed outputs.

| Input(s)                                   | Output(s)  |  |
|--|--|--|
| 1. <i>Album Query</i><br>e.g. Lost in Love | a) Track list  | b) Album art   |
|  | c) Pricing information (per album)   | d) Listener ratings and reviews                                      |
|  | e) Available free options (streaming etc)  | f) Similar albums recommendations                                    |
|  | g) Music video of album’s hit song   |  |
| 2. <i>Song</i><br>e.g. The Prayer          | a) Lyrics  | b) Album art   |
|  | c) Pricing information (per song)  | d) Available free options ( <i>e.g.</i> , streaming, free downloads) |
|  | e) Similar songs recommendations   |  |
| 3. <i>Artist</i><br>e.g. Dionne Warwick    | a) Biography   | b) Discography   |
|  | c) Photos (from album art, or live concerts)   | d) Similar artistes recommendations                                  |
|  | e) *Upcoming events ( <i>e.g.</i> location, ticket prices, weather, airport flights) | f) *Collaborations   |
| * <i>optional features</i>                 |  |  |

*Table 1- iMusic Inputs/Outputs*

## 2.2 Functionality

iMusic will use several APIs along with custom built queries to facilitate the integration of data to enhance the overall user experience as it pertains to music search via iWisdm. To ensure the best results from iMusic, a survey will be conducted, where potential users will be asked to rank the given functionalities and also suggest possible ones. iMusic's functionalities include:

1. Accepting user queries
2. Providing lyrical, pricing, track list information
3. Providing latest news about artist/group
4. Displaying videos of an album's top hits
5. Refer to Table 1 above for other functionalities.

## 2.3 Architecture

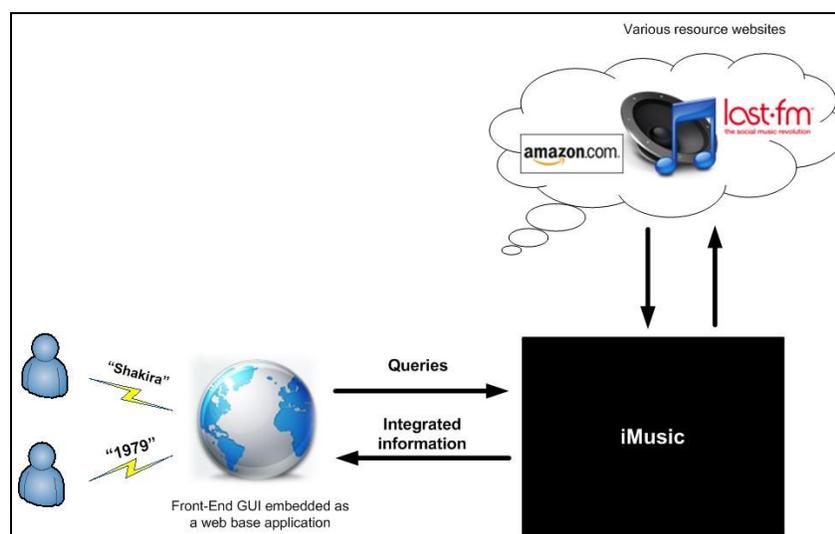


Figure 1- iMusic architecture

## 3. Implementation

The implementation will be done using the following APIs: a) amazon b) iTunes c) last.fm d) flickr e) YouTube and f) lyricsfly. We will also be scraping information from gracenote, and imeem.

## 4. Challenges

- **Disambiguation:** How to tell if the query is a song title or the name of an artist? Current music search services mistake a famous song title for a not so famous artist, and vice-versa. For example, "1979" is a famous song from "Smashing Pumpkins", but it's also the name of a Chinese DJ (<http://www.last.fm/music/1979>).
- **Usability:** What does a user want displayed, and what is the best way to portray the information?

## 5. References

[1] - <http://mostpopularkeywords.net/rank/?keyword=music>

[2] - Daniel Gruhl, Meena Nagarajan, Jan Pieper, Christine Robson, Amit Sheth. "Context and Domain Knowledge Enhanced Entity Spotting in Informal Text". *To appear in the Proceedings of the 8th International Semantic Web Conference (ISWC) 2009*. Fairfax, Virginia, USA. October 2009.