CSECS 2011, pp. 245 - 254

The 7 Annual International Conference on
Computer Science and Education in Computer Science,

July 06-10 2011, Sofia, Bulgaria

DIGITAL LIBRARY AND SEARCH ENGINE OF BULGARIAN FOLKLORE SONGS

Kiril KIROV, Nikolay KIROV

***Abstract**:* We present a full text search engine in a collection of lyrics (text of songs) and coded notes (symbolic melody) as a part of digital library of Bulgarian folklore songs.

***Keywords**: search engine, digital library*

***ACM Classification Keywords**: H3.3 Search process; J5 Literature, Music*

1. Introduction

The Bulgarian folklore music is a valuable resource of cultural heritage and is one of the main characteristics of the national identity of Bulgarian people. Throughout the XX century the Bulgarian researchers of musical folklore had written down hundreds of thousands of musical folklore songs in lyrics and notes. Part of these music notations have been published, another part is preserved as manuscripts in specialized institutional or personal archives. The major part from the available music notations with Bulgarian national music however is on paper. In the second half of XX century the researchers have recorded (usually on magnetic tape) performances of authentic singers of these songs. Today's information technologies give us the possibility to digitize the existing manuscripts and musical records [16]. The presented here search engine was implemented in Ruby programming language [1]. Its source code could be found at [15]. It can be used as a console or web application for keyword-based searching in a library of authentic folklore songs. The folklore songs are provided as an index of digital content – lyrics, notes and images. This engine could be used by professionals in the field of folklore research to look for common motives, characters and similarities between different folklore songs. These could be folklore songs from different parts of Bulgaria, different variants of the same song or simply common keywords.

2. Input Data Formats and Representation

The basic data files can be defined in five types:

- ➢ Ruby configuration file. The whole system must be configured from a Ruby config file. This file should contain the paths and file formats of all the content.

- LaTeX lyrics files. The lyrics of the songs in this library are written in LaTeX typesetting system [8]. In addition to the song text, each LaTeX file provides meta-data information encoded in the text. This meta-data is in the form of different LaTeX commands, and could be used both in compilation of the source and generating the index.
- LilyPond notes files. The notes of the songs are written in LilyPond music typesetting system [9].
- MP3 digitized authentic performance files. These performances are digitized from magnetic tape libraries of the Archives of Bulgarian Academy of Sciences. They had been recorded in different rural parts of Bulgaria during the 60s and 70s.
- JPEG digitized handwritten texts. The handwritten note-books included in this library were made by the experts, who worked with the authentic performers and then analyzed the collected data. They are valuable source of information about the circumstances and different traditions associated with the performance of each particular song.

Examples of these files can be found in [17].

2.1. The Search Engine Index

The system uses Ferret [10] – a high-performance, full-featured text search engine library written for Ruby. It is inspired by Apache Lucene [12] and implemented in C programming language. The `bin/index` command is used for building the search engine index, using the provided in the configuration file (see Configuration) data. The search engine index has a table structure, with different fields provided by the different types of input files. The input files for indexing are lyrics and notes. The lyrics files could contain meta-data, which is parsed and treated specially by the indexing engine. The meta-data is preserved in the text fields of the index table, and could be used to form search queries.

2.2. Configuration

The search engine config file is `lib/folk.rb`. It should contain the following data:

- Index path – the path to the index directory;
- LaTeX commands used for indexing – description of the LaTeX meta-data commands that should be used for building the index;
- Google Maps API key – obtained from Google API access key;
- Number of CPU cores of the system – rebuilding the whole library could be a slow process. The system could speed it up by using different CPU cores for parallel recompilation.

2.3. Compilationon

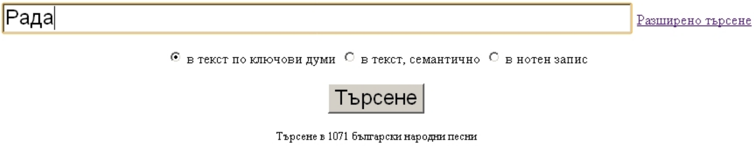
The compilation process could be started by `bin/lilypond` and `bin/latex`. It could take a very long time depending on the number of songs in the library, their complexity and the hardware parameters of the server. The compilation process generates the following output formats:

- LilyPond EPS and PDF engravings;
- LilyPond generated MIDI music;
- LaTeX EPS and PDF lyrics.

2.4. Searching

The search engine provides a Google-like web interface that would be used for searching in the library (Fig. 1). It uses a search phrase (query) that must be written in Ferret Query Language [11]. Search phrase could contain data as well as meta-data, as defined in the configuration file. A short list of possible searches:

- `Рада` – A simple one word search for “Рада” - a popular given name in Bulgarian. This search should return a result with all songs that contain that word. Note that “Рада” could be the name of the singer or the name of the folklore hero for whom the song is made.



Рада

● в текст по ключови думи ○ в текст, семантично ○ в нотен запис

Figure 1 Web interface for serching in the library

- `code:ba_002_2_04` – A code search. Every folklore song in the library has a unique code. The “code” is a separate field in the index table, so we specify a field using the shown above syntax.
- `content:"ожадня стоян за вода"` – A whole phrase search. Should return songs with containing the given words in the given order. “content” is a keyword for the lyrics field.
- `ст*ян AND area\{ямболско\}` – A wildcard and boolean search. In the different folklore songs the name “Стоян'” is sometimes spelled “Стуян”, so we want to match both of them. We also want to search only in the “ямболско” municipality, so we specify a meta-data field, which describes that area.
- `notes:fermata` – A note search. This search should return all of the songs, which contains a “fermata” (an element of musical notation) in their LilyPond coding.

3. Output Data

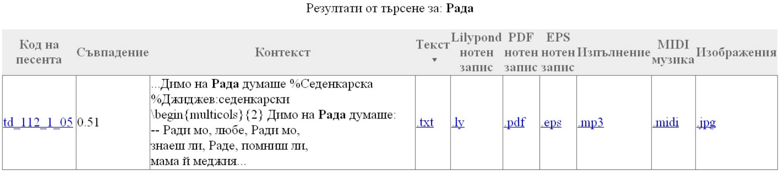
This system uses an integrated Ruby web server stack to server and present data in web form to its users. This stack includes the following Ruby Gems [2]:

- Thin – A web server [3].
- Rack – A web server interface [4].
- Sinatra – Web development framework [5].
- HAML – A web page template system [6].

The output data from the system could be summarized in the following two different categories.

3.1. Search Result Table

The search result table contains the songs that match a given search query. Each song is represented by a row in that table, which contains all the data and meta-data in the library about that song. Every song is identified by its unique code. By default the search result table is sorted by the relevance index given by the search engine. So the best matches are shown first. In addition to that the user could sort the table by any field. This happens in the web browser using JavaScript sort table script. The context of the given match is also displayed in the search result table, so the user could see the specific stanzas for example, that contain the given word. The user also could hear the authentic performance online, by clicking on the given MP3 link, for the specific song in the search results. A compiled, MIDI version of the Lilypond source file could also be heard. That could be used as a reference between written notes and the authentic performance.

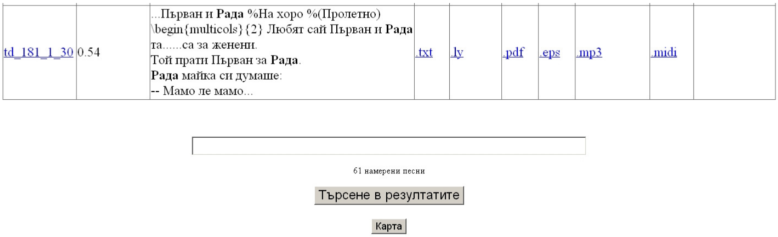


Резултати от търсене за: **Рада**

Код на песента	Съвпадение	Контекст	Текст ▾	Lilypond нотен запис	PDF нотен запис	EPS нотен запис	Изпълнение	MIDI музика	Изображения
td_112_1_05	0.51	...Димо на **Рада** думаше %Седенкарска %Джиджев:седенкарски \begin{multicols}{2} Димо на **Рада** думаше: -- Ради мо, любе, Ради мо, знаеш ли, Раде, помниш ли, мама й меджия...	.txt	.ly	.pdf	.eps	.mp3	.midi	.jpg

Figure 2 The top of search result table

Links to the lyrics, notes and images are also provided. Because of the specific Ferret Query Language grammar, forming exact search queries could be difficult. So the user has an option to form a louse query, which could match a large number of songs, just to see how it works. Then he can refine the query further and run a new search, but this time not in the whole library, but only in the results of the first, broader search.



td_181_1_30	0.54	...Първан и **Рада** %На хоро %(Пролетно) \begin{multicols}{2} Любят сай Първан и **Рада** та......са за женени. Той прати Първан за **Рада**. **Рада** майка си думаше: -- Мамо ле мамо...	.txt	.ly	.pdf	.eps	.mp3	.midi	

61 намерени песни

Figure 3 The bottom of search result table

3.2. Google Maps Visualization

In every step of the search process, a link is provided that could visualize the resulting songs in Google Maps [13]. The system extracts the relevant meta-data from the index and forms a series of Google Maps queries, that should return the exact location (or locations) associated with each given song. These queries are formed as strings containing the name of town or village, where the song was performed or associated with, and the municipality in which that town or village is located. Since Google uses keyword based search, that pair should be enough to distinguish between names of villages located in different municipalities (which is a common occurrence in Bulgaria). Google Maps queries return a GPS coordinates and a JavaScript based map, on which each song location is visualized. By using this technique a user could figure out how a given song motive is spread across rural areas of Bulgaria. It can be used also to track the locations associated with different singers.

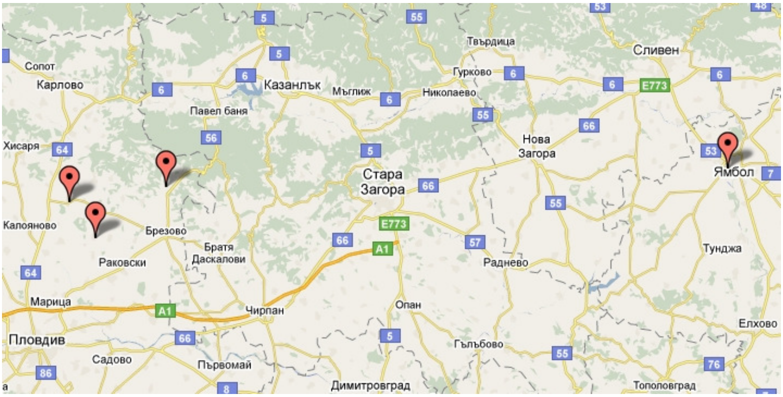


Figure 4 Google Maps visualization

4. Future Development

4.1. AST Analysis

During the LilyPond compilation, an Abstract Syntax Tree [14] is generated, base on the syntax of the input file. That syntax tree is interesting, because it provides all the necessary information for a possible automated song analysis and additional meta-data generation. Luckily the LilyPond provides an access to the AST by the means of a Scheme programming language API [7]. The system could provide a way to load an externally supplied Scheme script for automated analysis of the compiled LilyPond notes.

4.2. Usage on the Internet

In its current state the folklore songs search engine library is highly experimental and meant to be used only by experts and folklore professionals. The web interface of the system could be improved and redesigned according to web standards. Such improvement would make the system easier to use for professionals in the field and also usable to the interested communities on the Internet. The system is in no way limited specifically for Bulgarian folklore songs. It could be used for archiving, digitizing and indexing of different collections of authentic folklore art in the world at large.

Acknowledgement

This work is supported by Grant of the Bulgarian National Science Foundation under number DTK-02-54/2009 (see [17]).

Bibliography

[1] Ruby, Programming language, http://www.ruby-lang.org

[2] Ruby Gems, http://rubygems.org

[3] Thin, web server, http://code.macournoyer.com/thin

[4] Rack, web server interface, http://rack.rubyforge.org

[5] Sinatra, web framework, http://www.sinatrarb.com

[6] HAML, HTML templates, http://haml-lang.com

[7] The Scheme Programming Language, http://www.scheme.com/tspl3

[8] LaTeX typesetting system, http://www.latex-project.org

[9] LilyPond music engraving program, http://lilypond.org

[10] Ferret Search Engine, http://www.davebalmain.com/trac

[11] Ferret Query Language,

http://www.davebalmain.com/api/classes/Ferret/QueryParser.html

[12] Apache Lucene, http://lucene.apache.org/java/docs/index.html

[13] Google Maps, GIS, http://maps.google.com

[14] Abstract Syntax Tree, http://en.wikipedia.org/wiki/Abstract_syntax_tree

[15] GitHub https://github.com/kirilk/folk

[16] L. Peycheva, N. Kirov, M. Nisheva-Pavlova, Information Technologies for Presentation of Bulgarian Folk Songs with Music, Notes and Text in a Digital Library, Proc. of Fourth Int. Conf. "Information Systems & Grid Technologies", Sofia, Bulgaria, May 28–29, 2010, 218-224.

[17] Information technologies for presentation of Bulgarian folk songs with music, notes and text in a digital library,

http://math.bas.bg/or/nkirov/2010/folk/FOLK_en.html

Authors' Information

Your photo here:
Height: 2,58 cm
Width: 1,84 cm

Kiril KIROV, Magrathea Ltd.,Druzhba, block 405, vh.G, ap.75, 1582 Sofia, Bulgaria, kiril@kirov.be.

Major Fields of Scientific Research: *Programming, Databases, Information Systems, Digital Libraries*

Your photo here:
Height: 2,58 cm
Width: 1,84 cm

Nikolay KIROV, PhD, Assoc. Prof., New Bulgarian University, 21 Montevideo str., Building 2, office: 611, 1618 Sofia, nkirov@nbu.bg and Institute of Mathematics and Informatics, BAS, Akad. G. Bonchev str., block 8, 1113 Sofia, Bulgaria, nkirov@math.bas.bg.

Major Fields of Scientific Research: *Digitization, Databases and Programming, Astronformatics*