Marzena MIĘSIKOWSKA
KIELCE UNIVERSITY OF TECHNOLOGY, FACULTY OF ELECTRICAL ENGINEERING, AUTOMATICS AND COMPUTER SCIENCE

Speech command based application enabling Internet navigation

Mgr inż. Marzena MIĘSIKOWSKA

Assistant in the Department of Computer Science, Faculty of Electrical Engineering, Automatics and Computer Science, Kielce University of Technology. Research interest: digital signal processing, designing and managing database systems.

e-mail: marzena@tu.kielce.pl

Abstract

Thie paper presents an attempt to create an application enabling the user to surf much easier the resources of the Internet with the help of voice commands, as well as to classify and arrange the browsed information. The application has two basic modules which enable browsing the information on the Internet. The first navigation module processes websites, isolates navigation elements , such as links to other websites, from them and gives an identification name to the elements, which enables the user to pronounce voice commands. The website is presented to the user in a practically original form. The second module also processes websites, isolating navigation elements from them. The only difference in operation of the both modules is the mode of processing the website and its final presentation. The second module isolates from the elements vocabulary, which makes it possible to classify the information included in the website, this way acquiring and displaying, an ordered set of navigation elements. The application was implemented in Java language with the use of Oracle software. For the system of recognition and understanding of speech the Sphinx 4 tool was used [1].

Keywords: speech recognition, text classification, information retrieval.

Aplikacja umożliwiająca nawigację w Internecie za pomocą poleceń mowy

Streszczenie

W tej pracy przedstawiono próbę stworzenia aplikacji umożliwiającej swobodniejszą nawigację użytkownika wśród zasobów Internetu za pomocą poleceń mowy, klasyfikację oraz uporządkowanie przeglądanej informacji. Aplikacja posiada dwa zasadnicze moduły, przy pomocy których możliwe jest przeglądanie informacji w Internecie. Pierwszy moduł nawigacji, przetwarza strony internetowe, wyodrębnia z nich elementy nawigacyjne takie jak odnośniki do innych stron, oraz nadaje elementom identyfikacyjną nazwę, dzięki której użytkownik może wydawać słowne polecenia. Strona internetowa wyświetlona zostaje użytkownikowi w niemalże oryginalnej postaci. Drugi moduł również przetwarza strony internetowe, wyodrębniając z nich elementy nawigacyjne. Jedyną różnicą w działaniu obu modułów jest sposób przetwarzania strony i ostatecznej jej reprezentacji. Drugi moduł wyodrębnia z elementów słownictwo, dzięki któremu możemy sklasyfikować informację znajdującą się na stronie, uzyskując i wyświetlając w ten sposób uporządkowany zbiór elementów nawigacyjnych. Aplikacja zaimplementowana została w języku Java z wykorzystaniem oprogramowania Oracle. W przypadku systemu rozpoznawania mowy zastosowano narzędzie Sphinx-4 [1].

Słowa kluczowe: rozpoznawanie mowy, klasyfikacja tekstu, przeglądanie informacji.

1. Introduction

The paper presents an application by means of which the user can browse through the Internet using voice commands. The application presented enables controlling a browser by voice commands, as well as with the use of a keyboard or mouse. Design of applications that support voice commands is becoming more and more important. It is caused by the appearance of better and better systems of recognizing human speech.

Two techniques are used for looking through the Internet resources. First of them is so-called browsing – following Internet links. The second way is so-called querying – usage of Internet browsers [3]. The created navigation module uses the first way of browsing through the resources, following the navigation elements, made available by various Internet resources.

The contemporary Internet portals are complex. Often, the information presented is not ordered. Taking these facts into consideration, the application was equipped with a module classifying navigation elements. The classification aims at accelerating the process of obtaining the desired information. The next sections of the paper contain detailed information on the application. Motivation for creating the application is presented in Section 2. Section 3 describes the architecture and functioning of the application. The results are given in Section 5, while Section 6 presents the concluding remarks.

2. Motivation

The main factor for creating the application was the amazingly rapid development of the Internet. The Internet is a unique information resource for individual users, companies and institutions. An Internet browser is an absolute minimum for looking through the Internet resources. There are numerous applications available that enable us to surf the Internet. Most of them are controlled by means of a keyboard and mouse. The environment of such applications is friendly only for a particular group of users.

More and more designers of *open source* applications use the mechanisms of recognizing speech [4, 5]. The operation of such applications includes mostly creating editors of documents.

The applications equipped with a module of recognizing human speech are called *user-friendly applications*. Of course it does not only concern the users, who to some extend have problems using the keyboard. Using the keyboard and listening to the commands from the microphone at the same time, since such a way of working is possible, may help to perform some tasks faster.

Each project of application must fulfill some requirements. As far as this application is concerned, the quality of recognizing and understanding human speech, as well as the constraints connected with recognition, are of high importance.

There can be distinguished the support of the significant majority of the English language phonemes. It should be noted that in some cases of particular actions connected with voice commands of the user, the training of reception and the ability to cooperate with packets of programs and the operational system are essential. So, the next motivating factor was the attempt to meet the requirements made for this type of applications.

3. Application – architecture and functioning

The functioning of the application, i.e. the way in which the particular modules exchange information, is illustrated in Fig. 1.

Commands pronounced by the user to the microphone are transferred to the speech recognition module, Sphinx-4. It recognizes the commands supported by the application and sends them to the navigation module which fulfills the task and sends the results to the user. The speech recognition module is ready to listen to next commands of the user.

The application architecture is presented in Fig. 2. In the design of architecture, the pattern of command was very helpful. The feature of this pattern is sending commands to the objects of

commands. The objects of commands use methods of the receivers of commands to answer the command.

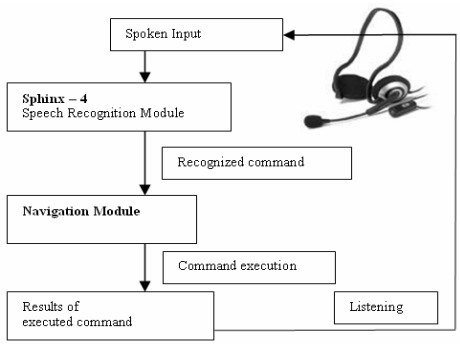

Fig. 1. Information exchange between modules
Rys. 1. Sposób wymiany informacji między modułami

The application was implemented in the Java language, therefore the architecture presented in Figure 2 is a facilitated diagram of classes. All the elements of the diagram were made by the author. The diagram does not include the architecture of the speech recognition system Sphinx-4.

More information related to this system architecture can be found in [1].

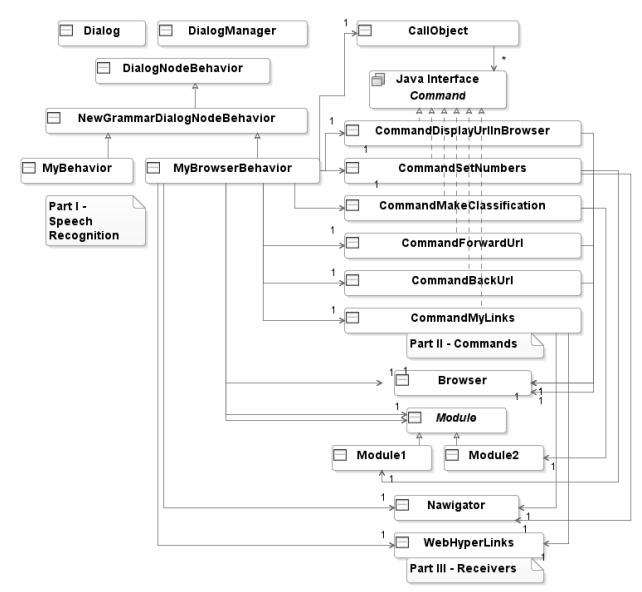

Fig. 2. Application architecture
Rys. 2. Architektura aplikacji

The application architecture consists of three basic parts:

- Part one – the module responsible for listening to the commands of the user. In order to recognize speech properly, this part of application employs the system Sphinx-4.
- Part two – the activating object, stores the objects of the command and, in a proper moment it manages them in order to carry out particular commands of the user.
- Part three – the receivers, executors of commands. The objects of commands form a bound between operations and the executors of commands. The calling object sets a command to be carried out by calling a proper method of the object of command. The object of command carries out the command by calling proper methods of the receiver.

The main task of the application is to launch particular actions, connected with the commands pronounced by the user. The module of command recognition listens to the words pronounced by the user to the microphone, next, having recognized particular commands defined in the application it takes particular actions. The main object of part one is *MyBrowserBahaviour*. This object defines particular objects of commands and receivers of commands. It uses the *CallObject* to call the corresponding objects of commands in order to fulfill the task required by the user.

The commands of the user that the application can fulfill with the help of its receivers, according to the above architecture are:

- *Go to page {goto_page}* – moving to the mode of browsing through the Internet resources.
- *My links {my_links}* – the command causes the user's favourite links to appear in the navigator.
- *One – Five hundred* – numerals by means of which the websites corresponding to navigation elements in the browser can be opened.
- *Set numbers {set_numbers}* – displaying the navigation elements singled out on the website in the navigator
- *Make classification {make_classification}* – displaying in the browser the singled out and then arranged navigation elements.
- *Back {back}* – displaying the last watched site in the browser. The command corresponds to the command *back* of any Internet browser.
- *Forward {forward}* - displaying the next site in the browser.

Due to the availability of this application for most users and lack of unique names identifying navigation elements, it was assumed that particular navigation elements placed on the currently displayed site would be defined by means of numerals. Each site has a navigation map consisting of numerals and addresses corresponding to them.

A Call Object, after recognizing a particular command by the module of speech recognition, displays a particular object of a command to fulfill the command defined by the user. The object of the command fulfills the command by calling the proper methods of the receiver. According to the application architecture the following objects of command correspond to voice commands:

- *CommandMyLinks* – command *my links*. The command calls the methods of the object *WebHyperLinks* to read navigation elements. Next, with the use of the method of object, the *Navigator* displays the elements read in the navigator.
- *CommandSetNumbers* – command *set numbers*. The command calls proper methods of the object *Module1, Browser*, *Navigator* to read the contents of the Internet browser, identify navigation elements, build a navigation map and display the numbers and addresses in the navigator.
- *CommandMakeClassification* – command *make classification*. The command singles out navigation elements from the website by means of the methods of the objects *Browser* and *Module2*, classifies the elements and displays the ordered navigation elements in the Internet browser.
- *CommandDisplayUrlInBrowser* – command for numerals. The command uses the object *Browser* to follow Internet addresses corresponding to numerals, displaying websites that are placed at the particular address in the browser.
- *CommandBackUrl* – command *back*. The command uses the object *Browser* to display the previous site displayed by the user.
- *CommandForwardUrl* – command *forward.* The command by means of the object *Browser* displays the next site.

Commands performed by the receivers of commands. The defined receivers according to the application architecture are:

- *Browser* – the most important object – Internet browser. Methods made available by the Internet browser serve for displaying and browsing through the resources of the Internet.
- *Navigator* - graphic navigation panel – it has a graphic panel similar to the browser, serving first of all to display navigation elements and numerals corresponding to them, achieved by the command *set numbers*. It consists of two tabs. The first tab displays the favourite links of the user, while the second – navigation elements of the site.
- *Module* [1, 2] – navigation modules. The first module reads the contents of the website, it singles out navigation elements and builds a map of internet addresses, which enables the user to browse through the resources of the Internet by pronouncing particular numerals. The second module also reads the content of the website and singles out navigation elements. Unlike the first

one, it singles out key words from navigation elements and then it classifies the elements into their corresponding key words. This way, the second results in a sequence of navigation elements.

- *WebHyperLinks* – an object responsible for reading and saving to a file the navigation elements defined as often used by the user.

Fig. 3. A fragment of Navigator
Rys. 3. Fragment nawigatora

The application enables the user to browse through websites in two ways: displaying the website in its natural form or displaying it with ordered navigation elements. The browser provides approximate options, present in every Internet browser.

4. Implementation

The application was implemented in the programmers' language Java with the use of Oracle JDeveloper software. Sphinx4 software was used [1] with the module of speech recognition.

Front-end of the application is presented in Figure 4. It consists of an Internet browser and navigator. The browser displays the website in its original form. If the user chooses the mode of classification of navigation elements, then instead of the original website, the ordered navigation elements will be displayed. The navigator will display the navigation elements of the original website.

Fig. 4. A Browser with Navigator
Rys. 4. Przeglądarka z nawigatorem

The application was implemented in such a way that it allows at the same time navigation by means of voice commands as well as a keyboard and mouse. To display the content of a given Internet address in the browser, the user should read the number placed in the navigator's toolbar. After displaying the website, in order to browse through its contents, the user should use the command SET NUMBERS.

The displayed ordered set of navigation elements is obtained by pronouncing the command MAKE CLASSIFICATION. The navigator consists of two tabs: tab1 and tab2. In order to move to the first tab, the user should pronounce the command MY LINKS.

The user may use a casual browser available for his operational system. The advantage of the system Sphinx-4 is the ability to work on many system platforms.

5. Results

The application performs the basic task of Internet navigation, i.e. it enables the user to browse through the Internet resources by means both of voice commands and a keyboard and mouse. The application may be equipped with more commands because it has at its disposal a very good module of recognizing speech. Sphinx4 it able to recognize speech without any previous training conducted by the user.

The application cooperates with the operational system. Websites may be viewed in the system browser of the user. The cooperation with the operational system, with the use of the browser supplied together with the operational system was tested on the basis of the platform of Windows XP Professional.

The classification module is recommended when the site is so huge that finding a particular information takes a long time. Arranging elements on the website increases the possibility of finding particular information when the user is performing a different task at the same time.

6. Concluding remarks

The paper presents a model of application, which makes it possible to navigate through the Internet with the use of voice commands. An Internet browser is not only controlled by voice commands, but also by a keyboard and mouse.

Applications controlled by means of voice commands are more available and friendly to the majority of users.

The earlier attempts of voice navigation resulted in receiving a wide range of users' experiences during the work with such a system. The increase in the utility and better understanding of the demands of the users is necessary for formulating assumptions, working out alternative designs and, in general, for a more profound understanding of the problem and its possible solutions. The advantages of such an approach are: faster completing the work on the system and involving the user into the construction of the system.

7. References

[1] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, Joe Woelfel: Sphinx-4: A Flexible Open Source Framework for Speech Recognition, SMLI TR-2004-139 Sun Microsystems Inc., November 2004.

[2] Skubis T., Dulas J.: Parametryzacja sygnału stochastycznego za pomocą siatek dwuwymiarowych, Pomiary Automatyka Kontrola nr 7/8, 2002.

[3] A. Abdollahzadeh Barfourosh, H.R. Motahary Nezhad, M. L. Anderson, D. Perlis: Information Retrieval on the WWW and Active Logic: A Survey and Problem Definition, 2002.

[4] Michel Généreux, Alexandra Klein and Harald Trost: A Multimodal Speech Interface for Accessing Web Pages, Conference TALN 2000, Lausanne, 16-18 October 2000.

[5] Shairaj Shaik, Raymond Corvin, Rajesh Sudarsan, Faizan Javed, Qasim Ijaz, Suman Roychoudhury, Jeff Gray, Barrett Bryant: Speech-Clipse – An Eclipse Speech Plug-in, Eclipse Technology eXchange Workshop (OOPSLA), Anaheim, CA, October 2003.

[6] Basztura Czesław: Komputerowe systemy diagnostyki akustycznej, Wydawnictwo Naukowe PWN, Warszawa 1996.

[7] The Source for Java Technology Collaboration, http://java.net/

[8] Pascale Fung, Cheung Chi Shun, Lam Kwok Leung, Liu Wai Kat and Lo Yuen Yee: Salsa Version 1.0: A Speech-based Web Browser for Hong Kong English, 5th International Conference on Spoken Language Processing (ICSLP 98), Sydney, Australia, 1998.

[9] K. Huang and J. Picone: Internet-Accessible Speech Recognition Technology, presented at the IEEE Midwest Symposium on Circuits and Systems, Tulsa, Oklahoma, USA, August 2002.

[10] Dulas J.: Zastosowanie metody siatek o zmiennych parametrach do identyfikacji fonemów mowy polskiej, XL VIII Otwarte Seminarium z Akustyki, 2001

Artykuł recenzowany