Voice based Internet Browser

Gobbur Uday Pune University Dept. of Comp. Engg. P.R.E.C Loni. Kale Ketan Pune University Dept. of Comp. Engg. P.R.E.C Loni.

ABSTRACT

Websites are accessible only through devices equipped with a screen and a Graphical User Interface (GUI). This requires physical interaction with devices. We present a system which allows browsing the Internet by using a standard voice only, with the development of a Vocal User Interface (VUI). The system accepts vocal commands as input from the user, translates those commands into HTTP requests, sends them to the web server which processes it and finally returns the HTTP response translated back to the user in a vocally manner. To reach this goal the system implements Content Extraction (CE) algorithms over web content in order to analyze, classify and return relevant parts of web pages to the user.

The Microsoft speech SDK allows applets transmitted over the Internet or intranets to access speech capabilities on the user's machine. This provides the ability to enhance World Wide Web sites with speech and support new ways of browsing. Speech recognition can be used to control browsers, fill out forms, control applets and enhance the WWW/Internet experience in many other ways. Speech synthesis can be used to bring web pages alive, inform users of the progress of applets, and dramatically improve browsing time by reducing the amount of audio sent across the Internet.

Its applications are numerous: for example helping the blind access the internet through speech and hearing, helping disabled people or young children unable to use a keyboard to "speak" their commands into the web, or, simply enabling any person to interface the web contents via oral commands, instead of a keyboard.

Keyword: Content extraction, Voice reorganization

1. INTRODUCTION

Now a day, web browser is become a most popular sourse of information .Millions of people used it for their daily activities. The browser which are currently available are generally GUI browser. Generally browser are not able to operate by using voice command, because of that the people who are physically handicapped are not able to use this information source. This paper gives idea about handling of the complete browser only through voice. Using vocal access to the network one can access the complete browser. Different speech conversion methods are available such as voiceXML, java speech API, Microsoft speech Software Development Toolkit..Our system based on platform of Microsoft speech SDK(Software Development Kit) for conversion of voice to text. Umbarkar Dipak Pune University Dept. of Comp. Engg. P.R.E.C Loni. Neharkar Swapnil Pune University Dept. of Comp. Engg. P.R.E.C Loni.

This system is divided in to the various module. The modules like speech conversion, database, command checker, web page handler, system handler are the different modules of the system.

2. SYSTEM ARCHITECTURE

First of all the user give command to the system by the use of microphone. These commands are then pass to system tool called 'Microsoft speech SDK'. This tool converts the speech into text string. This string is input to our module which we going to be attached to the browser. This module consist of

- 1) Command checker
- 2) Browser handler
- 3) Web page handler
- 4) System updater.

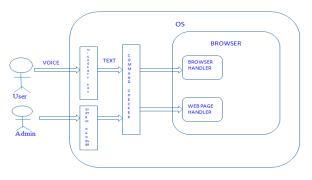


Fig 1: system Architecture

1) Command checker

We are maintaining the database of commands in the MSaccess. This module compares the string with the commands and if it matches then that command will send either to the browser handler or to the web page handler. If that command is not present then it will show you the error message.

Suppose, if command is like this 'Google dot com' then it will automatically converts that command like 'google.com'

2) Browser handler

The browser handler deals with actual browser commands like save, open, edit, copy, paste, etc. So here it just calls the system '.exe' file of that command, it will get open.

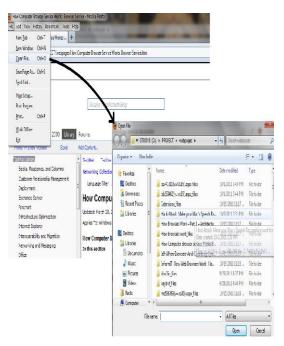


Fig 2: Browser Handler

3) Web page handler

The web page handler deals with web pages and performing the browsing activity like 'google.com' go to any website and search any content, access links etc. So here it takes the link of any hyperlink text form that source code present on page. That link gets gets copy to address bar of browser.

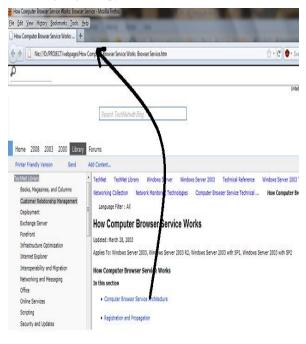


Fig 3: Web page Handler

if we want to click on 'computer browser service architecture' then the link present for it in source code gets copied in address bar and page gets open.

4) System updater

This is on the side of administrator. If he wants to make any changes in the database then he can use the system updater to change commands or add some more commands.

3. SYSTEM FUNCTIONALITY

For this project we are going to maintain the separate database. The command which we are going to predefined will be stored in this database. Also the website which user uses regularly will be stored in the database.

In the database, at least three table will be maintain. One table will be having the set of browser command with its appropriate action. The browser command just like save, download etc. That action will be invoke when the user speak that command.

The second table will be of website addresses. This table will having the URL of the website. When the user will speak just "open google" in this case google will be match with the www.google.com and this address will be put in the address bar to open that website.

This table will be updated when there will be new website found. That address will be inserted in the table and for the easy browsing we will provide the short command for that website so that it will be easier to open that website next time. Ex- for website like www.facebook.com the command will be ' facebook ' only.

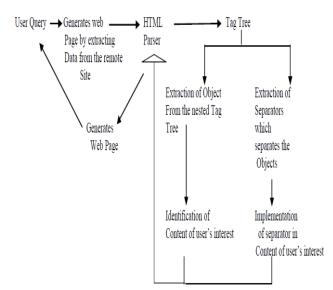
The third table will be belonging to the web page handling. This table will be created for the each webpage to handle the link on that web pages. This table will contain web pages link with the corresponding href. When the user gives the command of webpage link like "gmail" (link from the google.com) then corresponding link will be found out in the table and that href will be loaded in the frame. The temporary data structure will be created while matching the command and getting the links from the web pages.

4. USER INTERFACE

A vocal interface differs from a visual interface in that spokenprompts are transient and don't have any consistency. Furthermore, speech recognition is not 100% accurate. As a result, voice interfaces are much more dialog oriented, with emphasis on verbal presentation and response. Visual interface can present to the user more information at once, whereas a vocal interface can listen to the user only few amounts of data at a time. Therefore we had to design an efficient vocal interface pattern.. Once command accepted we have to execute them and also have to show the result to the user.

5. ALGORITHM

A Heuristic Approach for Web Content Extraction. The technique used in this paper works on analyzing the content and the structure of the web page. To extract the content from the web page; first of all we pass the web page through an HTML Parser. This HTML Parser is an open source and creates a tag tree representation for the web page. The HTML Parser used in our approach can parse both the HTML web pages and the XML web pages. Since the tag tree developed by the parser is easily editable and can be constructed back into the complete web page, the user gets the original look and feel of the web page. After the tag tree generation, the extraction of objects and separators are extracted using heuristics to identify the content of users interest.





Webpage before content extraction



Webpage after content extraction

6. CONCLUSION

In this paper we proposed a system able to translate visual websites into vocal applications giving the user a realistic browsing experience. Our system is based on the collaboration of distributed modules. Thus, it represents a significant progress in the world of Internet access and will generate new vocal applications based on richer vocal interfaces.

7. ACKNOWLEDGMENT

The authors would like to thank Prof.Jondhale S.D. from the Pravara College of Engineering for her valuable input and Prof. Korade for his supervision of part of this project.

8. REFERENCES

- Michael sazbon, yoram haddad, "advanced vocal web browser", school of engineering computer science & networks dept. Jerusalem college of technology (JCT), 2010 IEEE 26-th convention of electrical and electronics engineers in israel
- [2] Nang nay zar lwin, "agent based web browser", university of computer studies, mandalay (UCSM), myanmar, 2009 fifth international conference on autonomic and autonomous systems.
- [3] A. Saoudi L.I.P.N," Parallel Parsing of Context-free Text- Structure Grammars" Institut Galilke UniversitC Paris XI11 Avenue J. B. Clkment 93400 Ville t aneuse, France.
- [4] James R. Evans," Achieving A Hands-Free Computer Interface Using Voice Recognition And Speech Synthesis" Software Engineering Division (Tisad) ,Ogden Air Logistics Center,Hill Air Force Base, Utah 84056.
- [5] C.Chandramouli and Vivek Agarwal, "Speech Recognition based Computer Keyboard Replacement for the Quadriplegics, Paraplegics, Paralytics and Amputees" Applied Power Electronics Laboratory, Department of Electrical Engineering, Indian Institute of Technology -Bombay, Powai, Mumbai-400076, India.
- [6] Neha Gupta and Dr. Saba Hilal ," A Heuristic Approach For Web Content Extraction", International Journal of Computer Applications (0975 8887), Volume 15- No 5, FEB 2011