ISSN: 0975-766X
CODEN: IJPTFI
Available Online through *Research Article*
www.ijptonline.com

AN APPLICATION FOR WEB NAVIGATION USING SPEECH ENGINE FOR BLIND PEOPLE

Prof Vijayan R*
Asst. Professor (SG), School of Information Technology and Engineering, VIT University, Vellore-632014.
Email: rvijayan@vit.ac.in

Received on 25-07-2016 *Accepted on 28-08-2016*

Abstract

The World Wide Web today has become a major force in driving the economy and the technology. Most of the content in the web are in the format unsuitable for the disabled people. The main objective of the system is to access web in mobiles by voice feedback which has become a boon for the visually impaired people. Hence the system provides the blind users to navigate the web by voice browsers. As it is difficult for a visually handicapped person to identify the different links and forms present in the page, navigation through the page or through different pages are difficult for them. To go to the interested area he has to spend more time and has to scroll the entire page. The existing systems are now suffering with high power consumptions and processing time.DOM/SAX parser is used to extract Content of the web page. This parser also helps to restructure the content by eliminating flash ads. The user can just give the keyword of a link to open corresponding web page. The browser is integrated with speech engines. Hence the browser provides interactive voice user interface. E-Speaking is the speech recognition system is used. Verbose text-to-speech engine is used in our system.

Keywords: Text-speech conversion, Speech-text conversion, Parser, DTD parser.

1. Introduction

Internet has brought about an incredible improvement in human access to knowledge and information. However, blind people face difficulties in accessing these text materials. Hence architecture of web browser is designed in such a way that allows a blind person to navigate any web content through simple speech commands. This browser uses most efficient speech engines and a text extraction mechanism which is more accurate. This browser provides easy web navigation for a blind person. web browser has companion applications that "speech enable" it, allowing users to speak the name of a link and have the browser jump to that link as though it were clicked. This browser provides easy and fast web searching

mechanisms for blind people by using cheaper and efficient speech engines. This browser is integrated with efficient text extraction mechanism which consumes user time while searching. The main objective of the browser is to provide fast web searching mechanisms for blind people by using cheaper and efficient speech engines. DOM/SAX parser is used for the web content extraction consumes user time while searching. The parser eliminates flash ads and extracts only the main content. The main motive of developing the browser is to overcome the inability of a blind person in accessing the web content for provision of:

- To help the blind person to navigate web without any assistance.
- The browser is designed in such a way that he just need to tell the keyword of a link to open it.
- E-Speaking speech recognition will easily recognise the user voice and performs various tasks for him
- Verbose Text-To-Speech engine[6] can load any format of a file and converts written text to voice feedback for the user.

2. Related Works

Shruti Text-To-Speech [2] engine is used which translates only two Indian languages namely Hindi and Bengali. Gaussian smoothing[1] content extractor face some errors during extraction from web pages which contains long user comments and short primary content. DTD parsers supported in Stylus Studio include both DOM and SAX[8] based XML DTD Parsing components. A DOM used to builds an in-memory tree representation of the XML document. It provides classes and methods for an application to navigate and process the tree. An event-based API (SAX) uses calls to report parsing events to the application. The application deals with these events through customized event handlers. Events include the start and end of elements and characters.

- JERICHO parser[9] mainly focused on flexibility but not in the speed. The execution time for this parser will take more than DTD parser. Compared to tree based structure like DOM, the memory and resources requirements can be far better if only small sections of the document need to be parsed or modified. Compared to event based structure like SAX, the interface is much higher level.
- HTML and XHTML are two different way of representing that in mark up, but both are less expressive compared to DOM for example, "--" may be placed in comments in the DOM, but cannot be represented in a comment in either XHTML or HTML. The memory and time taken for HTML parser is more when compared with DOM/SAX parser.

- JSOP[10] is a java library for working with real-world HTML. It provides very convenient API for extracting and manipulating data. But execution time is higher when compared to DOM/SAX parser.

3. System Model

The problems in existing system here a system is proposed with Text-To-Speech engine which can load the text in all formats like pdf, Web page etc. and to increase the content extraction accuracy from the web pages containing short content but long comments summary of the special patterns will be given using DOM/SAX parser. The user gives voice commands using e-Speaking speech recognition software to open a web page. The web page is given as input to Text extraction engine.DOM/SAX parser retrieves the DTD events and forms tree structure. XSL-T processor stores parsed xml document. The recently visited url is stored in fie and will be given to the link extraction module. The retrived links and formatted text are stored in text files. Verbose Text-To-Speech engine loads the input file and gives the voice feedback to the user. The working of "DOM/SAX[1]" -Document Type Definition parser supported in Stylus Studio include both Document object model and Simple API for xml based xml dtd parsing components. When an xml file is passed through SAX parser there is an option on SAX interface to retrieve certain DTD events.DOM parser contains both methods and objects for navigating both xml and DTD Documents.DOM integrated with W3C DOM 1.0 Recommendation helps to access and manipulate XML Document as a tree structure in memory.DTD consists of several powerful tools and utilities includes DTD Editing, validation and conversion.XML documents are parsed using SAX integrated with SAX2.0 and SAX2-ext which uses event-driven model. Thus the xml document is parsed successfully. XSL-T processor includes many features like Built-in error recovery until fatal error and also support for JAXP 1.1.These are sent together with the parsed XML to the XSLT Processor where the selected style sheet is applied and the output is transformed (new) XML document.

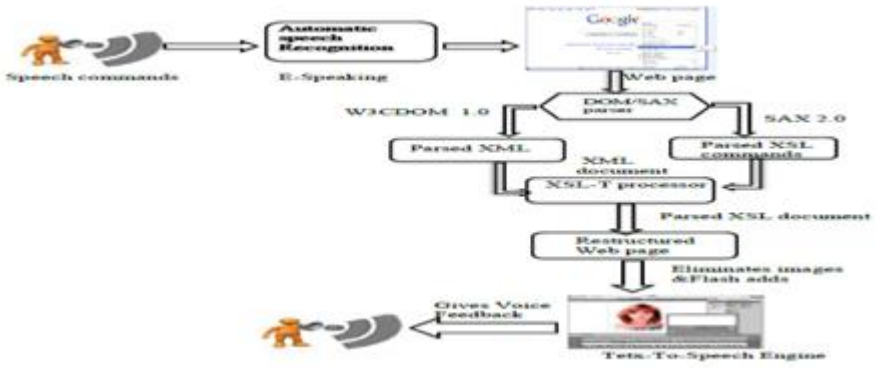


Figure 1: Proposed approach.

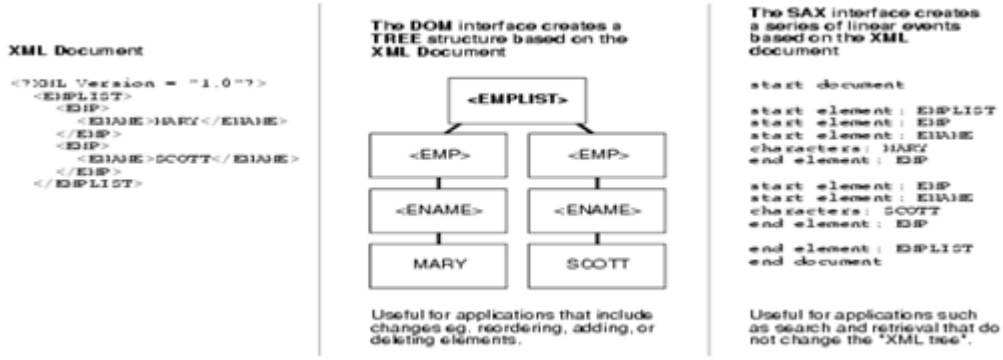


Figure 2: Construction of DOM Tree.

The working mechanism of the proposed browser:

- In the first step by using the speech recognition (e-speeking) the blind person will give the input in the form of speech commands. Speech recognition will convert the speech into text and opens the particular web page.
- The web page is parsed by speech recognition to the DOM/SAX parser.
- The DOM/SAX parser will remove the images and flash adds which are not useful to the blind person as described in the above session and give the restructured web page as output.
- After restructuring the web page we will give this as input to the Text-To-Speech engine which will read out the entire restructured page finally gives voice feed back to the blind person.
- to navigate through web it has been provided with keywords to the links. For each link the last words will be the key words.
- As the blind person is unable to remember the entire link it will be useful to provide the keywords. If he says the keyword through the speech recognition it will open the particular link.
- So that he can navigate through the web.

The proposed application is essentially composed of three modules namely User input module, Text extraction and re-structuring-Gaussians smoothing Content extraction and Output module

3.1 Input module

3.1.1 Automatic speech recognition

e-Speaking is the Speech Recognition Soft ware which converts speech commands into text.This software especially helps when surfing the web. Through small training sessions, the software learns how to recognize user voice. As time

progresses and the user can continue training the software, the accuracy and speed level will increase dramatically. It can be successfully utilized with internet explorer and mozilla firefox. This Software works three times faster than manually typing. The user can perform many tasks just by giving the voice commands. There are hundreds of built in commands. The user van add his/her own commands just by providing action part.The user can train any particular word so that it will perform much better. Hence this software helps the user to perform actions by just giving the voice commands.

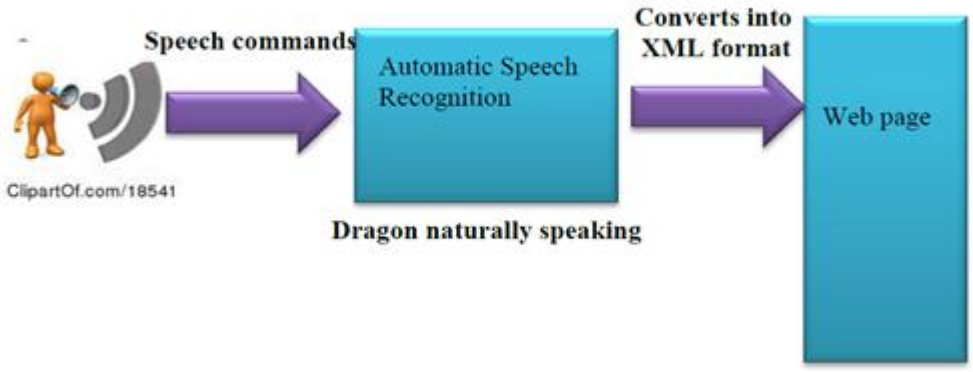


Figure 4. Input module of the system.

3.2 Text extraction and restructuring module

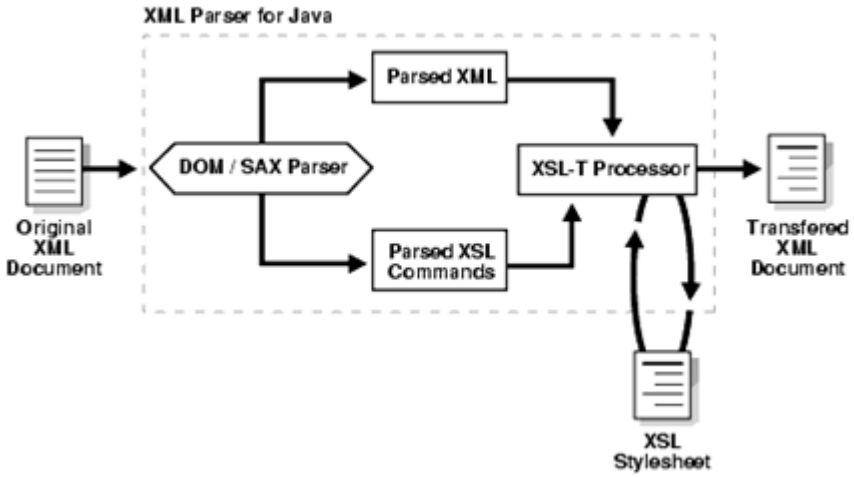


Figure 4: DOM/SAX Parser.

Document Type Definition parser supported in Stylus Studio include both Document object model and Simple API for xml based xml DTD parsing components. When an xml file is passed through SAX parser there is an option on SAX interface to retrieve certain DTD events.DOM parser contains both methods and objects for navigating both xml and DTD Documents.DOM integrated with W3C DOM 1.0 Recommendation helps to access and manipulate XML Document as a tree structure in memory.DTD consists of several powerful tools and utilities includes DTD Editing, validation and

conversion.XML documents are parsed using SAX integrated with SAX2.0 and SAX2-ext which uses event-driven model. Thus the xml document is parsed successfully.

XSL Transformation processor helps to transform xml documents from xml to xml or xml to any other text based format. XSL-T processor includes many features like Built-in error recovery until fatal error and also support for JAXP 1.1. If a style sheet is used, the DOM or SAX interface also parses and outputs the XSL commands. These are sent together with the parsed XML to the XSLT Processor where the selected style sheet is applied and the output is transformed (new) XML document.

3.3 Output module

Verbose converts text to speech more accurately and also more faster. The restructured text from the text extraction engine is finally sent as an input to the output representation module. Verbpse support multimodal input mechanisms and is also more reliable. Verbose Text-To-Speech engine will take any type of the input either in notepad, pdf or any kind of a web page. hence the speech output is given to the blind user which help him for web searching and can learn relevant material quickly consuming his time.

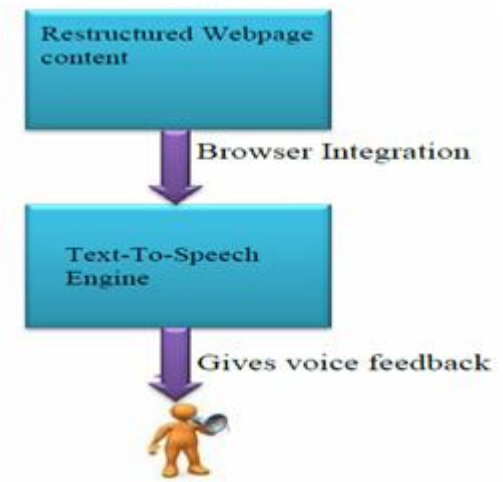


Figure 5: Text-To-Speech Engine.

The user gives voice commands using e-Speaking speech recognition software to open a web page. The web page is given as input to Text extraction engine. DOM/SAX parser retrieves the DTD events and forms tree structure. XSL-T processor stores parsed xml document. The recently visited url is stored in fie and will be given to the link extraction module. The retrieved links and formatted text are stored in text files. Verbose Text-To-Speech engine loads the input file and gives the voice feedback to the user. The Blind user gives the voice commands to perform any operation. E-Speaking recognizes

the commands and performs the actions and opens a web page. The web page is less formatted contains images, animations and flash adds. The web page is given to the Text extraction engine. It eliminates all the unwanted data and extracts only primary content. Hence the restructured data is given to Text-To-Speech engine and gives voice feedback to the user.

4. Implementation of the application

The current URL of a web page opened by the user will be automatically saved in IE history. The first URL in the history page will be copied into a file. The saved file will be given as input to the text extraction program. The history of recently visited web pages will be saved as follows from this the extraction of the first URL is done.

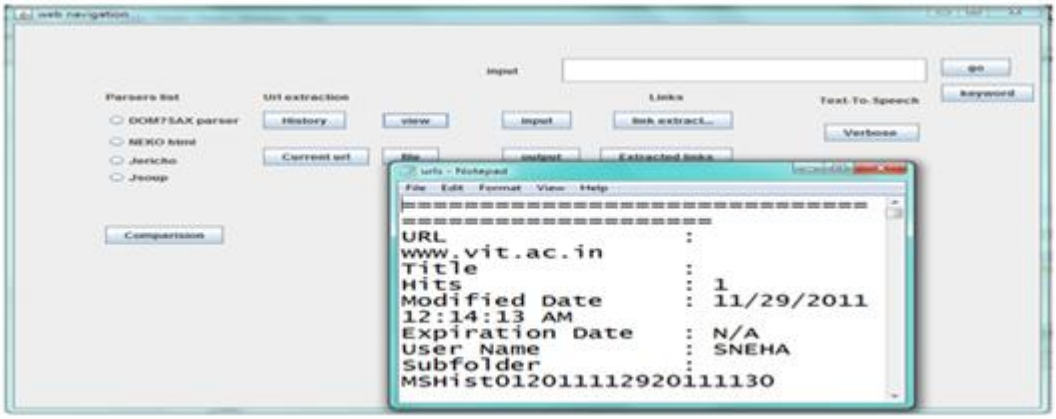


Figure 6: History Extraction.

This is the first URL present in the history of a browser. By using this blind person can easily navigate through web. In the proposed browser if the blind person clicks the button History it will automatically refresh the browsing history so that recently visited URL will be uploaded. Later if he clicks the button current URL the first URL from the history will be copied into a file. So that the user can extract the links

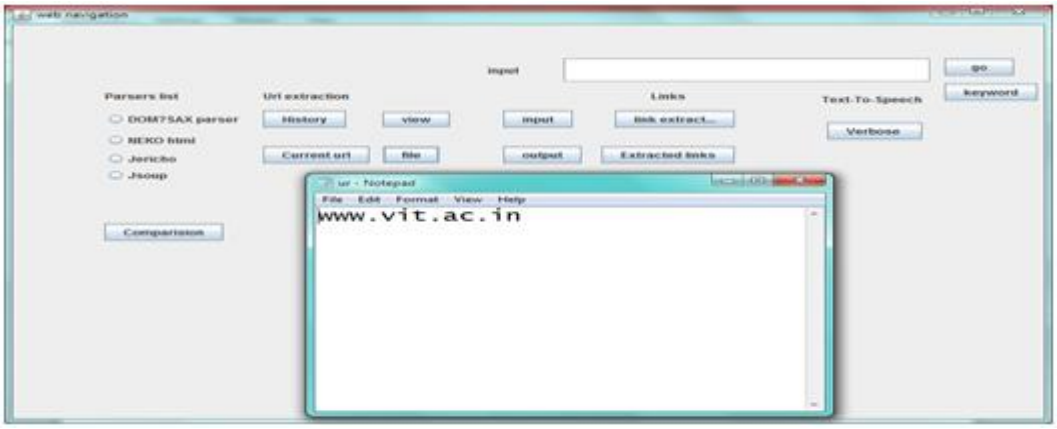


Figure 7: URL Extraction.

The implementation of the project is presented with the help of screen shots shown below. Each and every module of the project is tested as follows.It is also possible to extract the text from different formats of web pages like asp,htm etc. Here if this is link http://www.vit.ac.in is given as input for the browser and selected DOM/SAX parser from the list of parsers

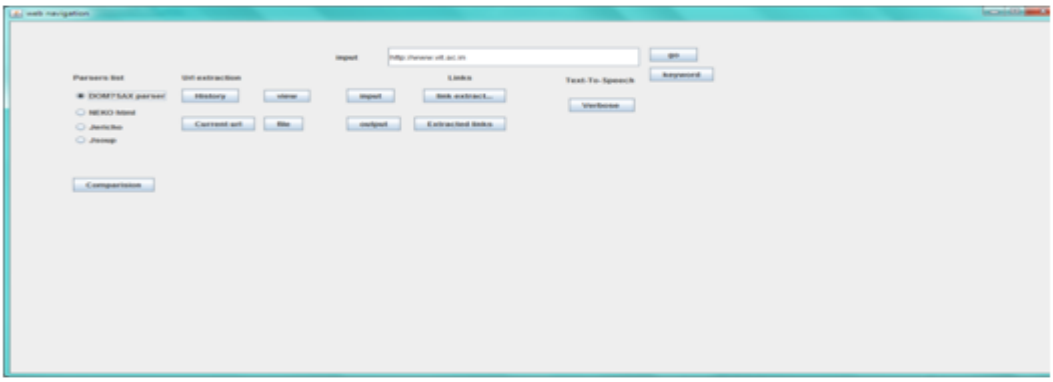


Figure 7(a): List of Parsers.

DOM/SAX parser finally eliminates images and flash ads as shown bellow and gives the main content

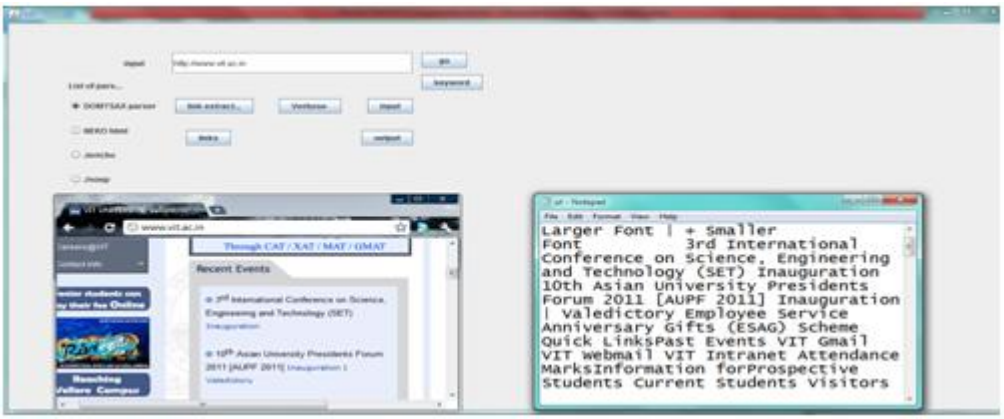


Figure 7(b): Text Extraction.

Hence the Text extraction is successfully parsed for asp formats. And the text file will be given as input to the Teext-To-Speech engine which gives the voice feed back to the user

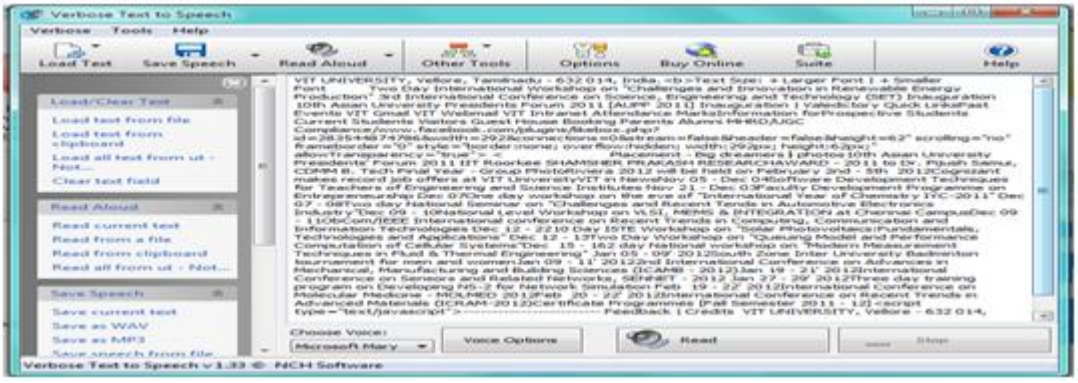


Figure 8: Text to Speech.

It is also possible to extract the text from html pages as shown bellow.For example for the link http://poi.apache.org/text-extraction.html the output will be as follows

Figure 9: Input the HTML page.

The corresponding text file will be created for the input URL as bellow

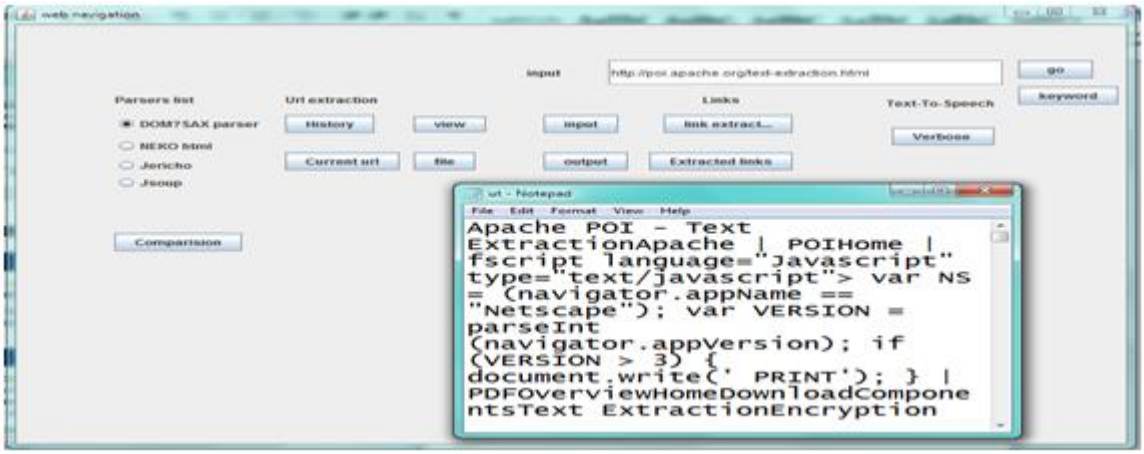


Figure 9: Text file creation.

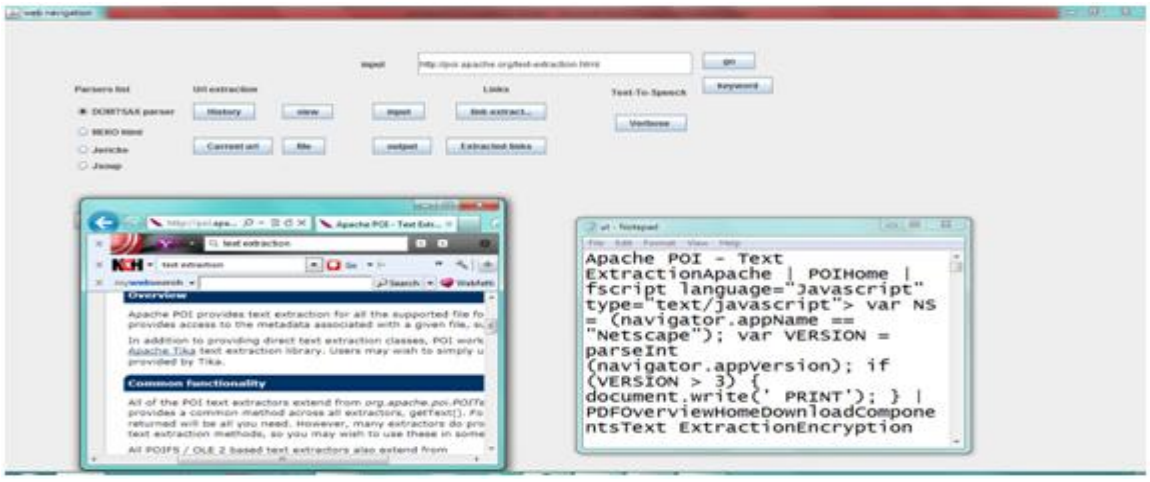


Figure 10: HTML Parsed to eliminate images and flash ads.

Hence the Text extraction is parsed for html pages successfully which will eliminate the images and flash adds and gives the main content for the users. Text extraction is also successful in extracting text from htm, php formats. Suppose give http://www.readwriteweb.com/hack/2011/03/text-extraction.php as input for text extraction.

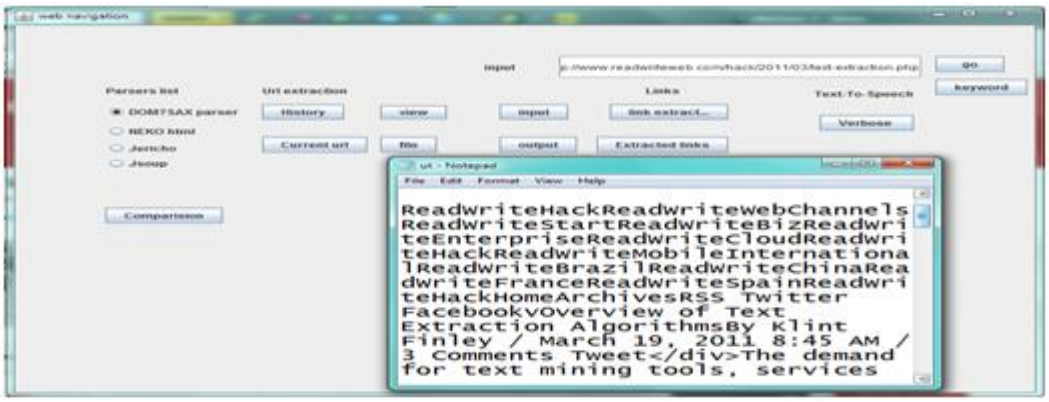


Figure 10: HTML Parsed to eliminate images and flash ads.

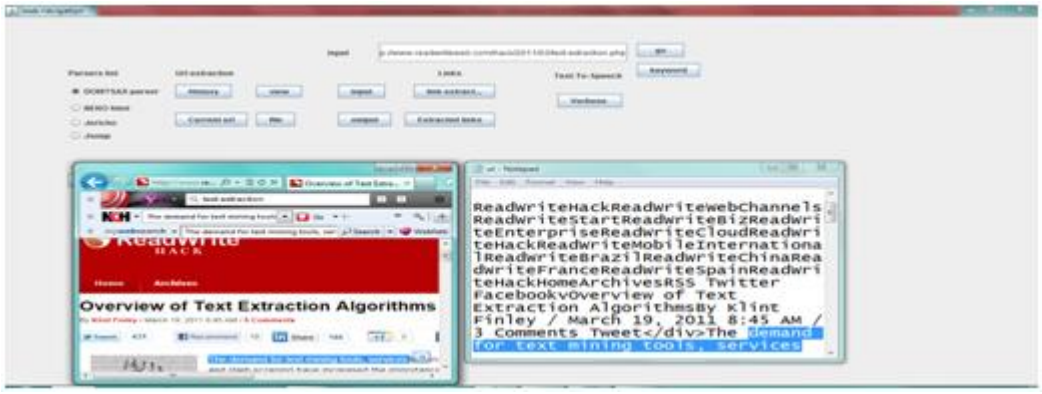


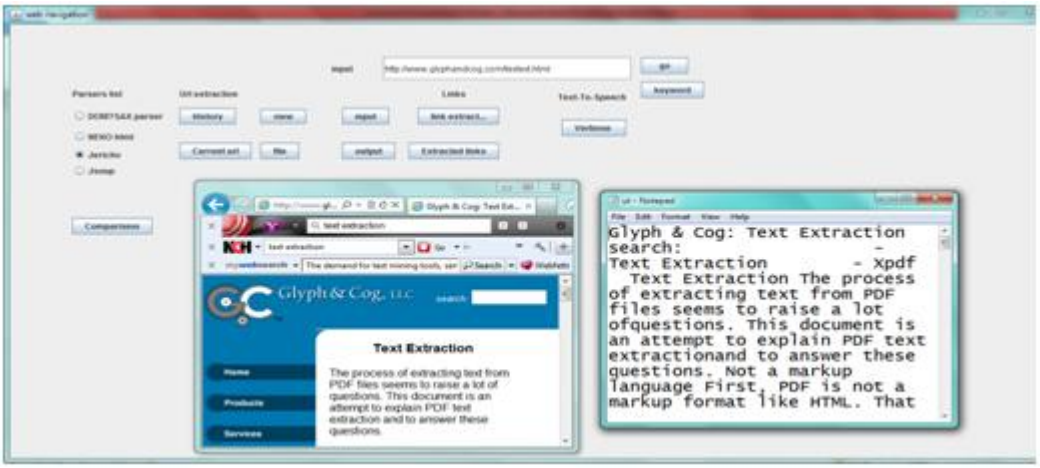
Figure 11: Text Extraction for PHP paes

Hence the text extraction for php pages parsed successfully as shown above figure 11.

In the proposed browser includes JERICHO, JSOUP and NEKO HTML parsers for text extraction. The text extraction for these parsers for the link will be as shown below

Figure 12: Text extraction for these parsers.

a)



b)

Figure 13: a) & b)Text extraction and restructuring for JERICHO parser.

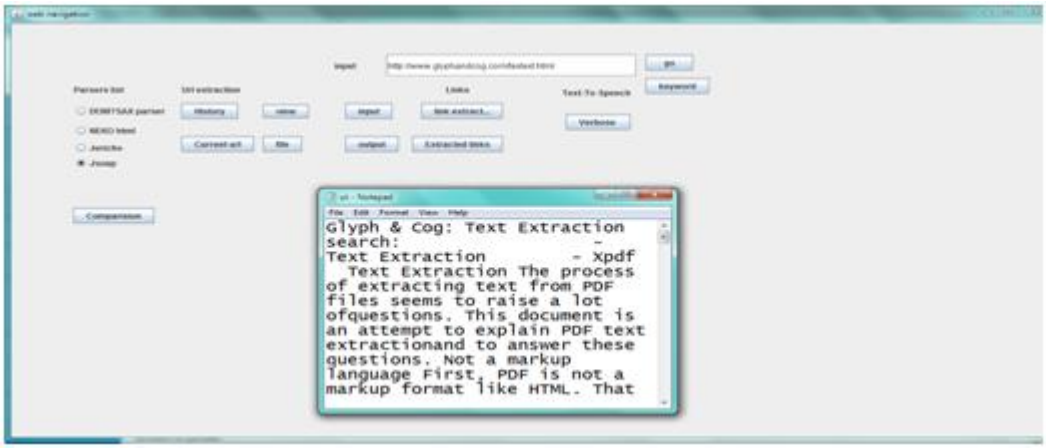


Figure 14: The text extraction for JSOUP.

Link extraction by using the proposed browser can also extract the links present in the particular web page and the corresponding keywords. To extract the links present in a web page are compared the tags with pattern <a.+href=\"(.+?)\" to easily extract the links. The user can also open the output text file by giving user commands. e-speaking used to perform this action. The corresponding text file for link extraction will be as follows

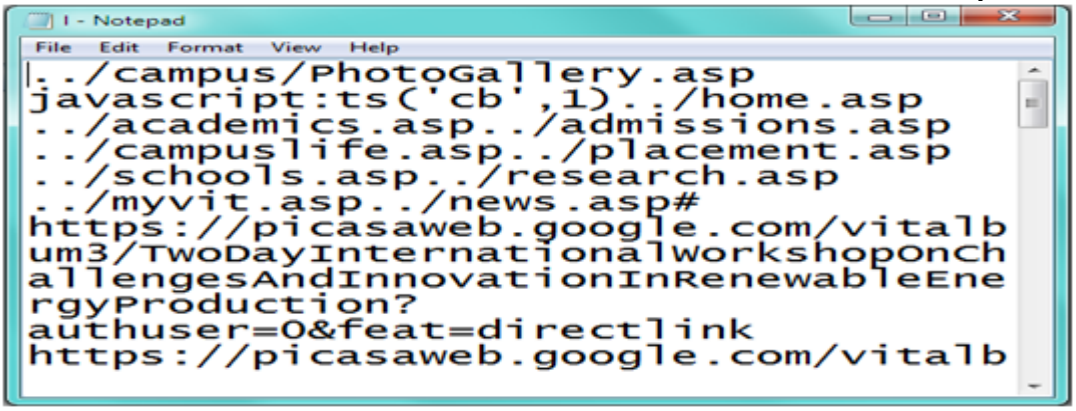


Figure 14: Text file for link extraction.

This text file will be given as input to the Text-To-Speech engine which is used to read the entire file and gives voice feed back to the user. The user has to load the text from the corresponding directory and then if he says “READ” it will give the feedback to the users, Loading text document and reading it as follows..

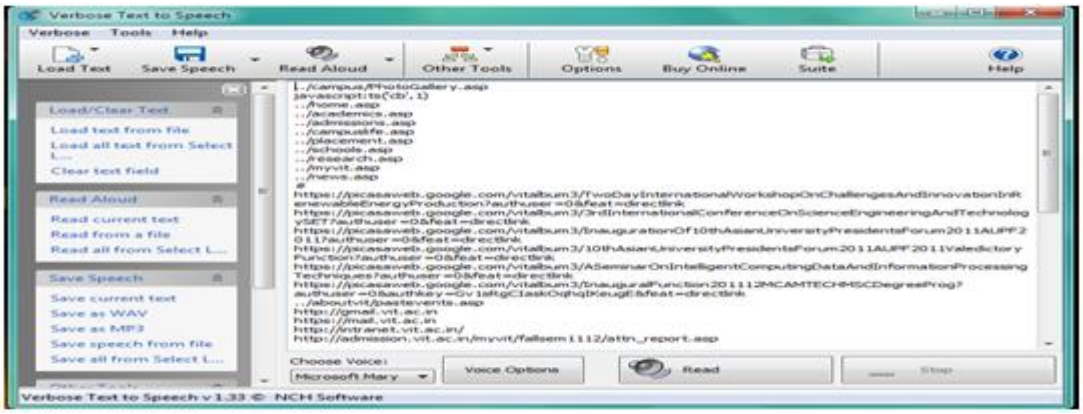


Figure 15: Links reading.

The last word of every link is the keyword. The keywords are stored in a text file. When the user gives any keyword it will search through the text file and its corresponding link will be opened. Hence it helps the blind person to navigate through the web

Figure 16: Apache POI

The corresponding links and its key words will be as follows

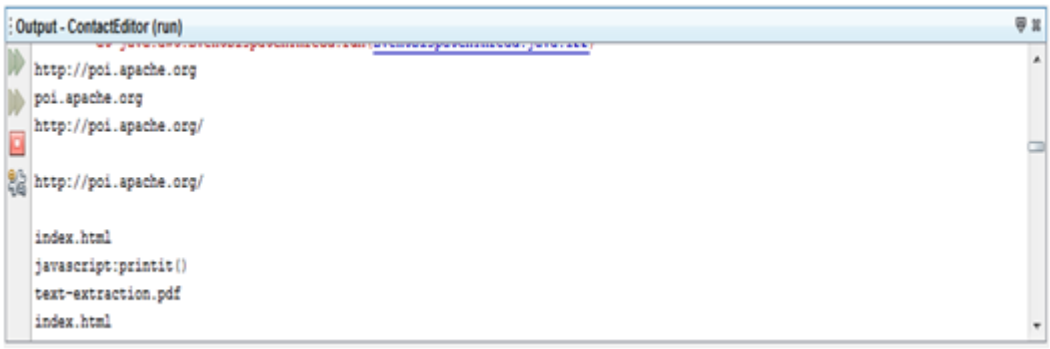


Figure 17: Links with keywords.

To the http://poi.apache.org link poi.apache.org will be the keyword. To open this link the blind person has to tell the key word then it will open the particular link in the web page.

Figure 17: Keyword to open links.

Here after entering the keyword it is opening the particular link as shown above. This is how the user can navigate through the web.

5. Results and Discussion

The below graph shows the comparison of various text extraction parsers based on the execution time of the algorithm.DTD parser is executed in less time than compared to other text extraction parsers.

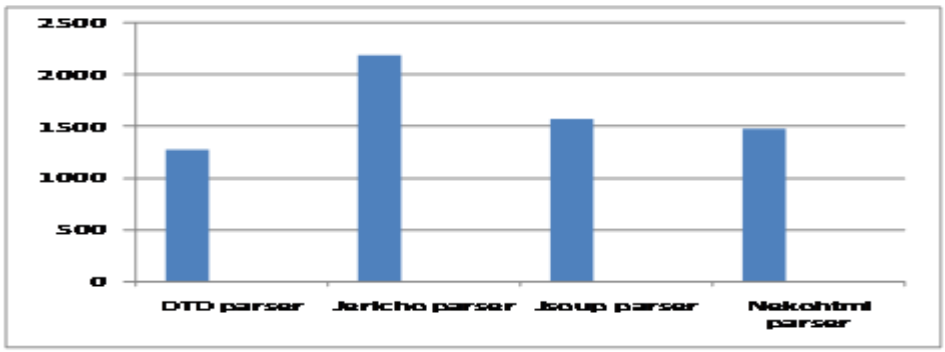


Figure 18: Comparison of execution time of various text extraction parsers.

Table 1: Comparison of various Text extraction parsers

Comparison	Methods used	Model used	Execution Time
DTD parser	DOM/SAX parser uses event driven model	Event driven model	Parser takes 1281 milliseconds to execute the algorithm
NEKO HTML[10]	Access html documents based on standard xml interfaces	Html scanner and tag balancer scans and fix errors in the html documents	Neko hmtl takes1481 millisecs to exeute the text extraction algorithm
JERICHO[9]	Uses simple but powerful java library for parsing html documents	Analysis of html forms to determine the structure of the submitted data	Jericho takes 2187 millisecs to complete the execution process
JSOUP[11]	Uses DOM,Jquery like methods to parse the html documents	Find and extract data using Dom traversal of css selectors	Jsoup takes 1578 millisecs to complete the execution process.

6. Conclusion

Thus the voice enabled web browser is successfully implemented. Browser is successfully integrated with e-Speaking speech recognition software and Verbose Text-To-Speech Engine. Hence the browser helps the blind person to navigate the web easily through voice commands. The Voice feedback is provided at every level which helps the user to navigate the web easily. The Browser also enables the user to just give the keyword to open any particular link. Hence it reduces the user effort in giving the entire link. Text-To-Braille conversion can be done as a future work. The Browser is expected to be more user friendly and hopefully reducing the gap between the sighted and visually impaired people

5. References

1. Ritwika Ghose, Tirthankar Dasgupta, Anupam Basu, "Architecture of Web Browser For Visually Handicapped People", Proceedings of the IEEE Students' Technology Symposium, pp 325-329, April 2010.
2. Halimah B.Z, Azlina A,Bchrang P, Choo W.O, "Voice Recognition System for the visually Impaired:Virtual Cognitive Approach", International Symposium on Information Technology, Vol 2,pp 1-6,August 2008.

3. L.B.W Zhang and H.Wang. "Primary content extraction with mountain model". In Proc of the IEEE CIT2008, pp 479-484,July 2008
4. K.R. McKeown, R.Barzilay, D.Evans,V.Hatzivassiloglou, M.Y.Kan,B.Schiffman and S.Teufel."Columbia multi-Document Summerization: Approach And Evaluation", In documenting understanding Conf, pp 1-21, 2001.
5. Text Extraction-http://poi.apache.org/text-extraction.html.
6. Verbose Text to Speech- http://www.nch.com.au/verbose/
7. DOM parser https://developer.mozilla.org/en/docs/Web/API/DOMParser
8. SAX parser-http://www.saxproject.org/
9. JERICHO parser- http://jericho.htmlparser.net/docs/index.html
10. NEKOHTML parser-http://nekohtml.sourceforge.net/
11. JSOUP parser-https://jsoup.org/

Corresponding Author

Prof.Vijayan R*,

Email: *rvijayan@vit.ac.in*