Text-to-Speech Technology-Based Programming Tool

DAT TRAN, PHILIP HAINES, WANLI MA, and DHARMENDRA SHARMA
School of Information Sciences and Engineering
University of Canberra
ACT 2601
AUSTRALIA

Abstract: - This paper presents an audio programming tool based on text-to-speech technology for blind and vision impaired people to learn programming. The tool can help users edit a program then compile, debug and run it. All of these stages are voice enabled. The programming language for evaluation is C# and the tool is developed in Visual Studio.NET. Evaluations have shown that the programming tool can help blind and vision impaired people implement software applications and achieve equality of access and opportunity in information technology education.

Key-Words: - Text-to-speech technology, C# programming language, Visual Studio.NET, Blind people

1 Introduction

According to the World Health Organization (WHO) globally, an estimated 40 to 45 million people are blind and 135 million have low vision [1]. In Australia over 480,000 Australians are vision impaired in both eyes, while over 50,000 are blind. This number is expected to increase to more than 87,000 people within 20 years [2].

Currently, there are screen reader tools such as JAWS [3], Brailliant Braille [4] and Window-Eyes Screen Reader [5]. However, the costs for these tools are high and there is no tool that integrates the environment for compiling and debugging programs. Furthermore, there is not enough assistance for helping them learn to program in the leading edge language C#.

Blind programmers could compete in the IT industry when infrastructure suited mainframes more [6]. These days, with all of computers in the workplace, graphical windows applications are far more common. This means that blind programmers are now at a competitive disadvantage in the workplace and require special tools to be productive.

Blind and vision impaired people require two things to become programmers. They need up to date knowledge of leading technology, and tools that meet their own requirements [7]. This affects employment levels for blind and low vision people. With the current unemployment rate for blind and vision impaired at almost 70%, which is over four times the national average, specialized tools could help a great deal of people [8].

Our research project is to design an audio programming tool that meets specific needs of blind and vision impaired people in learning C# programming language. There are different forms of visual impairment, some people are blind from birth or from a very early age, others lose their sight as a result of accidents, disease or some affects of medication [10]. Therefore we concentrate on text-to-speech technology and we assume that blind and vision impaired people are not hearing impaired. The text-to-speech technology is used to make all components in the programming tool voice enabled. Text and other graphics features such as control size, location, and color that a normal vision user can see on the screen will be spoken out by a speech synthesizer. This tool has opened a great possibility that allows blind and vision impaired users to become programmers in the future. Currently, blind and vision impaired people have little access to current tools and assistance required for them to learn programming languages. Our aim is to help them achieve equality of access and opportunity in information technology education that will ensure meaningful and equitable employment for their lives.

We have invited blind and vision impaired people to evaluate our programming tool. Evaluations have shown that the tool can help them design and implement programs effectively.

Our research project can potentially impact the lives of blind and low vision people. This coupled with the impending labor shortage, as the baby boomers retire, means that anything that can give blind people an opportunity to acquire practical, technical qualifications could greatly benefit blind people and the whole economy. A tool that teaches programming is also a programming tool and it can potentially give jobs to people who were previously unemployable.

Our research project will also impact software development companies, governments, and educational institutions to develop software packages, educational programs and policies that meet the needs of blind and vision impaired people.

2 Current Applications and Projects

Optical character recognition and text-to-speech technologies are currently used in software applications for blind and vision impaired people.

The first application is for reading books or newspapers. The optical character recognition technology is applied to scanners that scan text and read it aloud. Typical devices for this application are Extreme Reader [11], Ovation and SARA (Scanning and reading Appliance) [12] provide blind users access to printed and electronic materials. Those are converted from text to speech and read aloud. Kurzweil system scan documents, store in files, and convert those to audio output [13]. Furthermore, Optical Braille Recognition (OBR) allows a user to scan a Braille page and convert it in to text [14]. This is a Windows software application to retrieve information that can be presented as the text used in all types of Windows applications. The Braille information in a small letter can be retrieved into computer form in the same easy way.

For reading text materials in computer, the most popular software for blind users is JAWS [3]. This software provides speech and Braille access to Windows operating system and applications including Internet Explorer without the need of special configurations. JAWS also provides a way to access Web pages.

A research project has been undertaken by Curtin University, Cisco Systems and the Western Australia Association [10]. The project is to identify tools and techniques appropriate for vision impaired students to study computer at tertiary level. This project recommends improvements included the need for professional development for lecturers and improved student access to electronic educational materials.

A computer education project recognized by Stockholm Challenge [15] is to reduce the digital divide and provide education and learning tools in digital format not available for the blind in Vietnam on paper support such as school books newspapers and reading material. This project aims to create a generation of blind computer users at different level nationwide, and to provide a community place to acquire computer skills and share information.

However, there is no existing software application designed to help blind and vision impaired people learn programming subjects in information technology and engineering. This motivates us to design and implement a simple yet efficient programming tool for blind and visual impaired users to develop software applications. In the next section, we will present our proposed programming tool and show how we can implement it. Testing and evaluation are also presented.

3 Proposed Audio Programming Tool

It is seen that the more formats of material people can access, the higher their employment opportunities are. There is a higher need for technical skills amongst people who are blind or have low vision. Blind people require supporting tools that meet their specific needs.

The programming tool is designed not only for blind users but also for vision impaired and normal vision users. The interface should be designed in a way that complies W3C standards for vision impaired users and should be user friendly. The programming tool should be able to help a blind user edit, save, compile, debug and run a program. Moreover, the tool should have program templates and intellisense (auto-completion) options for user convenience. In order to achieve these objectives, an iterative approach was used. Each part was developed, tested then improved upon and tested again. This meant that usability issues were always found and improved. The tool has been designed to provide voice for blind users and display suitable font, font sizes and color scheme for vision impaired and normal vision people.

3.1 Audio Code Editor

A user starts editing a program or loading an existing program using audio code editor. The program on the editor can be saved to a file or can be compiled, debugged and run. For each character entered, the code editor can speak it out. The user can use left, right, up and down arrow key to check any character in the program by voice. Some of key requirements for the code editor are as follows:

- Tell the user whenever it is loaded or activated.
- Ask the user's confirmation before it is closed; saving a file or opening a file.
- Tell the user the current line number.
- An option for the user to specify a line number and go to that line.
- Templates created in advance for every Console application and Windows application.
- Speaks all characters on a line of code.
- For Windows Applications, the user will design the graphical user interface by typing details (size, location, text, name, etc.) on the code editor. The code editor will convert details to C# code and place the code to a file.
- Allow the user to write C# code for event handlers.
- Help the user write code quickly and correctly by speaking out properties, classes, etc.

3.2 Audio Compiler and Debugger

The code compiler uses the C# software development toolkit (SDK) to compile the program. However, to have voice output, we add code for voice accordingly to the current program using a code modifier then use the C# SDK to compile the modified program. For Console application, adding code for voice can be performed by identifying code for text output then add code for voice accordingly. For Windows program, adding code for voice is more complex. Mouse and key event handlers will be added for the user to use mouse or keyboard to design a Windows form. Voice will be output when a control on the Windows form is focused to let the user know what the control is. The compiler also lets the user know if the compilation is successful or if there is a compiling error.

When there is a compiling error it then tells the user that there are compiling errors then reads out all the errors details, with the file name and line number. If the user presses predefined shortcut keys, it stops reading, jumps to that line in that file and reads that line to the user. The user can now fix the code and presses the combination key to hear the next error if any.

3.3 Audio Output

The code compiler uses the C# software development toolkit (SDK) to compile the program. However, to have voice output, we add code for voice accordingly to the program before it is compiled. This is done for any program that provides non-graphics or graphics output. Mouse or key event handlers will be added to provide audio output when the user moves the mouse over a control or presses the Tab key to focus on that control.

3.4 System Architecture

Figure 1 presents architectural design of the audio programming tool. C# and text-to-speech software development toolkits (SDK) are used. User can start a new project by choosing a template in a list of available templates. If the project is a Windows application, then the user can use the built-in GUI builder to create Windows controls by entering property values such as location, name, text, size, etc. When the user writes code, the built-in code auto-completer will help user write long class or method names.

When the user finishes the program and wants to compile and run it, the compiler will analyze the program and add code to produce voice accordingly. The modified program will be compiled and debugged. Errors if any will be output to a file and the speech SDK will read out an error at a time and guide the user to the line of code that contains the error in the program. This procedure will be repeated until there is no error in the program and the C# SDK will run it. Voice and text or graphics will be output and the user can use mouse or shortcut keys to check the outputs.

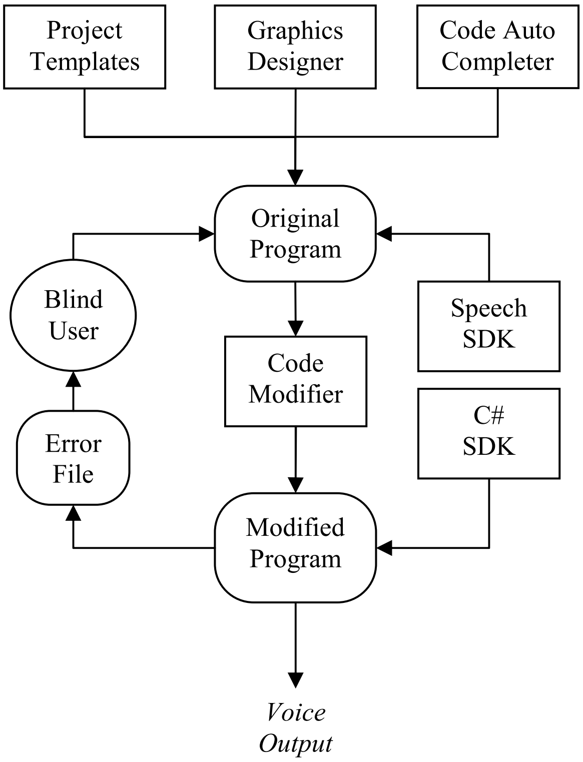

Fig. 1: The programming tool architecture

It is noted that if the blind user save the project to files and run it in the normal Visual Studio.NET, the output will be text or graphics only. Voice output is only available if the user runs the project in the audio Studio.NET.

4 Testing and Evaluation

The proposed audio programming tool has been tested and evaluated by normal vision users then by blind and vision impaired users. In the first test, normal vision users were required not watching the computer monitor when they tested the programming tool. It was observed that they were able to do all stages in writing a program by listening to voices output from the tool. In the second test, standard keyboards and built-in text-to-speech tools were used. We found that vision impaired and blind users were also able to perform the same task. However, vision impaired users were interested in applications with mouse and blind users prefer those with keyboard.

Most of blind and vision impaired people are familiar with shortcut keys defined in JAWS, so adding new shortcut keys in the programming tool is not recommended. Shortcut keys have been changed to meet their specific needs. More programming lessons need to be provided to help users be familiar with programming in .NET.

5 Conclusion

We have presented our design and implementation of an audio programming tool for blind and vision impaired people to learn programming in C#, a .NET language. The programming tool was designed not only for blind and vision impaired users but also for normal vision users. The programming tool was able to help a blind user edit, save, compile, debug and run a program. Moreover, the tool also had program templates and auto-completion options for user convenience.

The tool has opened a great possibility that allows blind and vision impaired users to become programmers in the future and to achieve equality of access and opportunity in information technology education that will ensure meaningful and equitable employment for their lives.

References:

[1] World Health Organization (2003). Retrieved from http://www.who.int/mediacentre/news/releases/2003/pr73/en/

[2] Access Economics (2004) Clear Insight: The Economic Impact and Cost of Vision Loss in Australia http://www.bca.org.au/natpol/statistics/

[3] JAWS (2007), retrieved from the following site http://www.freedomscientific.com/fs_products/software_jaws.asp

[4] Brailliant Braille (2007), retrieved from the site http://humanware.ca/web/en/p_OP_Brailliant.asp

[5] Window-Eyes Screen Reader http://www.tandt-consultancy.com/window_eyes.html

[6] Alexander Steve, (1998) Blind programmers face an uncertain future. Retrieved from the CNN: http://www.cnn.com/TECH/computing/9811/06/blindprog.idg/index.html

[7] Elkes, J. G. (1982) Designing Software for Blind Programmers. Public Utilities Commission of Ohio. Retrieved from an online article repository: http://delivery.acm.org/10.1145/970000/964173/p15-elkes.pdf?key1=964173&key2=4640659711&coll=GUIDE&dl=GUIDE&CFID=22945606&CFTOKEN=95515984.

[8] Vision Australia (2007) Results and Observations from Research into Employment Levels in Australia. Retrieved from the following site http://www.visionaustralia.org.au/docs/news_events/Employment_Overview.doc.

[9] Kopecek & Jergova (1998) Programming and visually impaired people, in Proceedings of ICCHP'98, Wien-Budapest.

[10] Ian Murray and Helen Amstrong (2004) "A Computing Education Vision for the Sight Impaired", in Proceedings of the sixth Australasian Computing Education Conference.

[11] Extreme Reader, retrieved from the following web site http://www.brailler.com/extrdr.htm

[12] Ovation and SARA, retrieved from the web site http://www.abledata.com/abledata.cfm?pageid=19327&ksectionid=19327&top=13293

[13] Kurzweil Education System. Retrieved from the web site http://www.kurzweiledu.com/

[14] Optical Braille Recognition, retrieved from the web site http://www.neovision.cz/prods/obr/

[15] Computer Education for blind people. Retrieved http://www.stockholmchallenge.se/data/computer_education_and_it