Assistive Interaction Techniques to Support Disabled Developers

Bharat Paudyal
DMT Lab, Birmingham City University
Birmingham
Bharat.paudyal@mail.bcu.ac.uk

Integrated Development Environments (IDEs) are widely used to support developers and heavily rely on using input methods (i.e. keyboard, mouse). Physically impaired developers who cannot utilise these traditional tools can therefore be excluded from development activities and prevented from entering into technical careers. Assistive tools such as speech recognition, eye gaze present an opportunity to address some of these challenges and make coding more accessible. However, they are often investigated independently with less emphasis around multimodal approaches. This dissertation therefore explores how these interaction methods can be integrated within an IDE to provide accessible coding approaches for physically impaired developers.

Eye gaze tracking, Speech recognition, Assistive technology, Software Development

1. INTRODUCTION

Software developers typically use Integrated Development Environments (IDEs) for application development as they includes useful features for syntactic highlighting, code navigation, autocompletion, debugging, etc. to help them not only write code, but also to manage large codebases (Albusays et al., 2017; Chi and Naik, 2019). The, majority of these IDEs require the use of traditional input devices (e.g. keyboard, mouse) as a method of interaction which might be inaccessible for developers with physical impairments (Hubbell et al. 2006; Rosenblatt et al. 2018; Delimarschi et al. 2014). Physically impaired developers with an interest in technical fields often face an arduous task when using these traditional input paradigms and are not always taken into consideration during the development of IDEs. This can result in them being excluded from technical fields and entering careers related to software development. Various assistive technologies such as eye gaze trackers, and speech recognition services have been developed and explored independently to support developers in writing code (Busjahn et al., 2015; Begel, 2005; Rosenblatt et al., 2018; Radevski et al., 2016; Glücker et al., 2014). Whilst these technologies can potentially support physically impaired developers, they each have limitations (e.g. speech recognition accuracy, selection of small interface targets via gaze input, etc.). To address these issues, this doctoral work is exploring how these methods can be utilized together within IDEs to present an enhanced approach to leverage the strengths of both technologies. This type of approach has the potential to provide a more viable and inclusive coding approach that can support professional level development work.

2. RESEARCH OBJECTIVES

The overarching aim of the research is to determine how assistive technologies such as eye gaze tracking and speech recognition can facilitate a usable, efficient, and feasible approach for performing coding activities. The research aim is to answer three core questions: (i) to what extent can the combination of speech input, gaze interaction, and mechanical switches make coding accessible for people with physical impairments, (ii) which are the optimal approaches for writing, editing, navigating, autocompleting code using these input technologies (iii) how can different speech interaction approaches for coding (i.e. natural language input, fixed vocal commands, speech label selections) facilitate coding activities. A series of individual studies, each having more specific research questions will be conducted that will focus around a specific element of coding.

Objective 1: *Exploration and Development of a Multimodal Inclusive Coding Environment*

Several studies have been carried out in non-programming domains (e.g. word processing (Beelders and Blignaut, 2010), web browsing (Sengupta et al., 2018), etc.) using the

combination of speech and gaze interaction. This work has commonly highlighted that the disadvantages of each modality can be reduced through using the two technologies in conjunction, thus making an enhanced interaction approach for people with physical impairments. To address the lack of work using this combined interaction technique in a development scenario, we want to combine eye gaze and voice as an approach for performing coding activities and evaluate the feasibility with both disabled and non-disabled users. In this work, we are developing a multimodal coding environment that will utilize speech and gaze to allow developers with physical impairments to perform coding activities such as writing code, navigation, deletion etc.

Objective 2: *Exploiting speech as a primary interaction approach to develop different voice coding methods*

Speech as an interaction approach has been explored to support developers in writing code, thus reducing the dependency on a traditional keyboard (Rosenblatt et al., 2018; Begel, 2005; Gordon, 2013). However, these tools are primarily targeted for experienced coders and a minority have been evaluated with physically impaired users. Additionally, they present limitations such as speech accuracy and having to learn new vocal grammars (Rosenblatt et al., 2018; Wagner and Gray, 2015). Hence, it is important to understand how developers verbalise code and which speech interaction approaches can be designed to facilitate coding activities for people with physical impairments. In this work, we will try to investigate how different speech interaction approaches (e.g. Natural Language Input, Fixed Grammars) can be developed and compared against accuracy and feasibility.

Objective 3: *Investigate code auto-completion and code navigation features using assistive technologies within a development environment*

Code autocompletion and navigation are important features in IDEs. In this work, we will focus on the interaction approaches being used when performing autocompletion and navigation by physically impaired developers in their current workflow. Based on the feedback from users, a speech and gaze based interaction approach will be developed and will be integrated within the coding environment. A usability study will then be conducted to investigate the feasibility of the developed prototype.

3. CONTRIBUTIONS SO FAR AND NEXT STEPS

A multimodal research prototype ("Voiceye") utilizing an eye gaze tracker (for typing code), speech recognition (for issuing commands to

control the interface) and mechanical switches (for selection) as a multimodal interaction approach for performing coding activities has been developed (figure 1). The prototypical code editor with simple design interface to existing mainstream development environments such as Atom, Brackets was developed using HTML, CSS, JavaScript, ElectronJS and codemirror, a JavaScript based open source code editor. The prototype consists of four components: Microsoft's Azure Speech Service as an automatic speech recogniser to convert speech into text, a rule-based grammar to trigger appropriate actions such as selecting, navigating, removing code based on the voice input, a gaze controlled onscreen keyboard to type code and two mechanical switches for selecting keys of onscreen keyboard and for triggering voice recogniser.
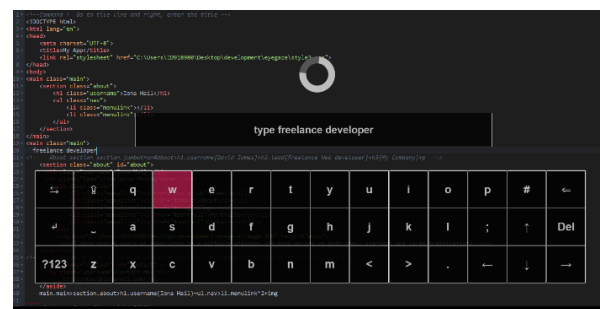

Figure 1: Screenshot of the Voiceye interface when the speech recognizer has been activated

This prototype was evaluated with both non-disabled (*N=29*) and disabled (*N=5*) participants to access the feasibility of writing HTML and CSS code as shown in figure 2. The design and evaluation were primarily focused on writing, deleting and selecting of the code.The results from two user evaluations were positive and demonstrated the feasibility of combining gaze, speech, and multiple mechanical switches interaction for supporting development work (Paudyal et al., 2020). The participants had the option of using two modalities in performing the coding activities and it allowed them to switch between the modalities based on their preferences.

Figure 2: A developer with physical impairment evaluating the Voiceye

The next stage of the work is exploring different approaches for writing code via speech interaction.

A focus group study has been conducted with able-bodied developers to obtain an understanding on how they might verbalise the writing of code in different scenarios. Based on the feedback collected, a voice based coding environment is currently being developed that will utilize different speech coding techniques (e.g. natural language input, utilising a fixed grammar, etc.). An evaluation will soon be conducted with both disabled and non-disabled users to compare the usability and feasibility of the developed coding approaches.

Following this work, the development of new techniques (using speech/gaze/switches) to support the process of autocompletion will be explored (to help make the code writing process for efficient for disabled developers). An online survey was conducted with 14 disabled developers to get an insights on the different code autocompletion approaches being used using different assistive tools. Drawing from the challenges found in the survey, different code autocompletion approaches will be developed and will be evaluated with disabled developers.

A further study will also investigate current code navigation practices and challenges faced by physically impaired developers and a prototype utilizing speech interaction will be designed to allow developers to perform code navigation.

4. RELATED PUBLICATION

5. ACKNOWLEDGMENTS

6. REFERENCES

Albusays, Khaled; Ludi, Stephanie & Huenerfauth, Matt (2017) Interviews and observation of blind software developers at work to understand code navigation challenges. *ASSETS 2017 - Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility* New York, USA. Association for Computing Machinery, Inc. 91–100.

Beelders, TR & Blignaut, PJ (2010) Using vision and voice to create a multimodal interface for Microsoft Word 2007. *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA '10* . NY, USA. ACM. 173.

Begel Andrew (2005) Programming by voice: A domain-specific application of speech recognition. *AVIOS speech technology symposium–SpeechTek. West.*

Busjahn, T et al. (2015) Eye Movements in Code Reading: Relaxing the Linear Order. *2015 IEEE 23rd International Conference on Program Comprehension.* 255–265.

Chi, Sheldon & Naik, Abdul Khader (2019) *An Empirical Study of How Developers Use Autocompletion.*

Delimarschi, Denis; Swartzendruber, George & Kagdi, Huzefa (2014) Enabling integrated development environments with natural user interface interactions. *Proceedings of the 22Nd International Conference on Program Comprehension*. New York. ACM. 126–129.

Glücker, Hartmut et al. (2014) EyeDE: Gaze-enhanced software development environments. *Conference on Human Factors in Computing Systems - Proceedings*. New York, USA. ACM. 1555–1560.

Gordon, Benjamin M (2013) Improving Spoken Programming Through Language Design and the Incorporation of Dynamic Context. (Doctoral dissertation, The University of New Mexico).

Hubbell, Thomas J; Langan, David D & Hain, Thomas F (2006) A voice-activated syntax-directed editor for manually disabled programmers. *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility - Assets '06*. 205.

Paudyal, Bharat et al. (2020) Voiceye: A Multimodal Inclusive Development Environment. *Proceedings of the 2020 ACM Designing Interactive Systems Conference* New York, USA. Association for Computing Machinery. 21–33.

Radevski, Stevche; Hata, Hideaki & Matsumoto, Kenichi (2016) EyeNav: gaze-based code navigation. *Proceedings of the 9th Nordic Conference on Human-Computer Interaction - NordiCHI '16*. New York, USA. ACM. 1–4

Rosenblatt, Lucas et al. (2018) Vocal Programming for People with Upper-Body Motor Impairments. *Proceedings of the Internet of Accessible Things.* 1-10.

Sengupta, Korok et al. (2018) Hands-free web browsing. *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications - ETRA '18*. New York, USA. ACM. 1–3.

Wagner, Amber & Gray, Jeff (2015) An Empirical Evaluation of a Vocal User Interface for

Programming by Voice. *International Journal of Information Technologies and Systems Approach* 8/2. 47–63.