Chapter 12
An Empirical Evaluation of a Vocal User Interface for Programming by Voice

Amber Wagner
University of Alabama, USA

Jeff Gray
University of Alabama, USA

ABSTRACT

Although Graphical User Interfaces (GUIs) often improve usability, individuals with physical disabilities may be unable to use a mouse and keyboard to navigate through a GUI-based application. In such situations, a Vocal User Interface (VUI) may be a viable alternative. Existing vocal tools (e.g., Vocal Joystick) can be integrated into software applications; however, integrating an assistive technology into a legacy application may require tedious and manual adaptation. Furthermore, the challenges are deeper for an application whose GUI changes dynamically (e.g., based on the context of the program) and evolves with each new application release. This paper provides a discussion of challenges observed while mapping a GUI to a VUI. The context of the authors' examples and evaluation are taken from Myna, which is the VUI that is mapped to the Scratch programming environment. Initial user studies on the effectiveness of Myna are also presented in the paper.

INTRODUCTION

GUIs can simplify user interaction by incorporating actions such as "drag and drop" or "click" to perform specified behaviors. The mechanism for this capability is often through the usage of a mouse requiring a minimum level of dexterity to navigate the GUI in addition to manipulating various objects on the screen. This required dexterity can pose challenges for users with physical disabilities. The ACM code of ethics states, "[i]n a fair society, all individuals would have equal opportunity to participate in, or benefit from, the use of computer resources regardless of race, sex, religion, age, disability, national origin or other such similar factors" (ACM, 1992). By not providing alternative means of access, users

DOI: 10.4018/978-1-5225-1759-7.ch012

with disabilities are denied opportunities, such as learning skills that may allow for the exploration of career paths in Computer Science/Information Technology. The recognition that more users should have the opportunity to use GUI-driven applications motivates the underlying theme of the work described in this paper.

As an example case study, Scratch (discussed further in the "Mapping Process Using Myna" section) is an Initial Programming Environment (IPE) using visual blocks as a programming metaphor. Scratch has a vast user base across a large age group and encourages a community of contributors where students can share and learn from each other (Scratch, 2015). Currently, there are nearly 8 million Scratch programs that have been shared among students and teachers worldwide (Scratch, 2015). As an initial study to show the potential for extending an IPE to address accessibility needs, we developed Myna (Wagner et al., 2012), which supports the concept of Programming by Voice (PBV) in Scratch.

The work presented here discusses challenges encountered when mapping a GUI to a VUI and evaluates the effectiveness of the VUI based on Wobbrock et al.'s (2011) definition of ability-based design. The remainder of this paper is organized as follows: Related Work discusses using voice as an input modality in a programming environment, and the subsequent section introduces Myna and discusses the mapping process (Mapping Process Using Myna). The User Testing of Myna section presents both a pilot study and a study with the target audience for this work (i.e., those with motor impairments who desire to learn about computer programming). Final thoughts are then summarized in the Conclusion.

RELATED WORK

GUIs typically require usage of the keyboard and mouse for input, which can be difficult for users with motor impairments. A suggested solution is to provide vocal input, which has proven to be successful in some cases (Begel, 2005; Dai et al., 2004; Désilets et al., 2006; Harada et al., 2009; Shaik et al., 2003). While many tools (e.g., Vocal Joystick) are advantageous and users find them beneficial (Harada et al., 2009), they may not be as helpful when manipulating objects within a GUI, such as in an IPE like Scratch (2015). Such programs require navigation in addition to manipulation of objects on the screen. Beyond providing a means for navigation, we believe that new accessibility tools should be created using ability-based design as described by Wobbrock et al. (2011), which is explained further in the rest of this section.

We believe that the key to creating a successful, universally usable tool is to apply an ability-based design, in which developers strive to take advantage of what abilities the user possesses and make the system adapt to the user, rather than require the user adapt to the system (Wobbrock et al., 2011). This design strategy should be utilized for systems/tools/applications for all users, not just those with disabilities. To ensure that a design is ability-based rather than disability-based, Wobbrock et al. (2011) developed seven principles: Ability, Accountability, Adaptation, Transparency, Performance, Context, and Commodity, which are explained in a later section in a discussion on how our evaluation suggests the degree to which Myna addresses these principles.

Martin et al. (1989) researched two claims regarding the validity of speech as an input modality: 1) Speech is faster than typing; and, 2) Speech increases productivity. Martin's literature search confirmed that using speech increases productivity and some of the research validated that speech is faster than typing, but some uncertainty remained. Martin created an experiment to test each claim by having users navigate a graphical application (called MAGIC). The resulting data confirmed both claims: speech had a 108% time advantage versus typing full-word commands, and speech increased productivity by allow-