THE UNIVERSITY OF ALABAMA COMPUTER SCIENCE

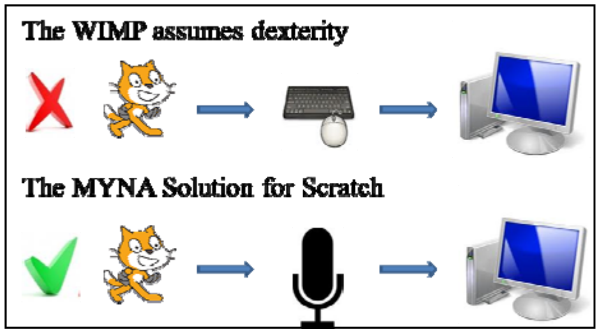
Programming by Voice: A Hands-Free Approach for Motorically Challenged Children

Amber Wagner, Jeff Gray
Dept. of Computer Science
University of Alabama
ankrug@bama.ua.edu
gray@cs.ua.edu

Ramaraju Rudraraju, Srinivasa Datla, Avishek Banerjee, Mandar Sudame
Dept. of CIS
University of Alabama at Birmingham
teammyna@googlegroups.com

Motivation

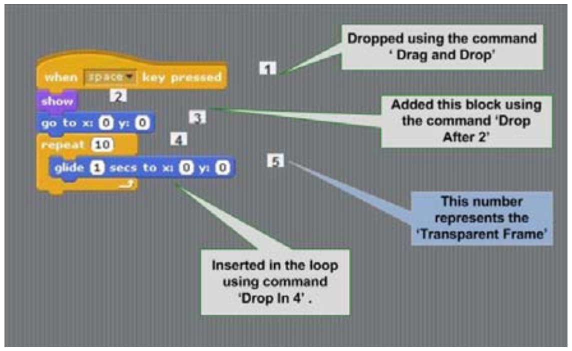
This poster introduces Myna, which assists those with restricted limb mobility in learning programming skills through a voice-driven interface. Specifically, Myna supports programming by voice with Scratch. Although the native Scratch environment allows users to create a program by arranging graphical blocks logically, Scratch is dependent on the Windows/Icons/Mouse/Pointer interface that requires dexterity in using a mouse/keyboard (limiting those with physical disabilities). Myna processes voice commands from the user, interprets those commands according to a Scratch-based grammar, and simulates synonymous actions of a mouse/keyboard in Scratch.



The result is an environment that assists those with disabilities (in particular, children) in experiencing the joy of programming.

Previous Work

To overcome the challenges of mapping the WIMP metaphor to a voice-driven interface, we investigated and developed continuous navigation and drag and drop capabilities, **Phase I**, illustrated in the following image.


when space key pressed
show
go to x: 0 y: 0
repeat 10
glide 1 secs to x: 0 y: 0
1
2
3
4
5
Dropped using the command ' Drag and Drop'
Added this block using the command 'Drop After 2'
This number represents the 'Transparent Frame'
Inserted in the loop using command 'Drop In 4'.

An important design consideration was the requirement that Myna exist outside of the Scratch implementation (i.e., Myna does not require any source code modification to the Squeak implementation of Scratch). This requirement posed several challenges (e.g., adding transparent frames as visual overlays to Scratch).

Myna Workflow

In this section, we demonstrate the flow of all parts of the architecture by considering a typical workflow of Myna. The following voice commands are mapped to the flow illustrated on Figure 3.

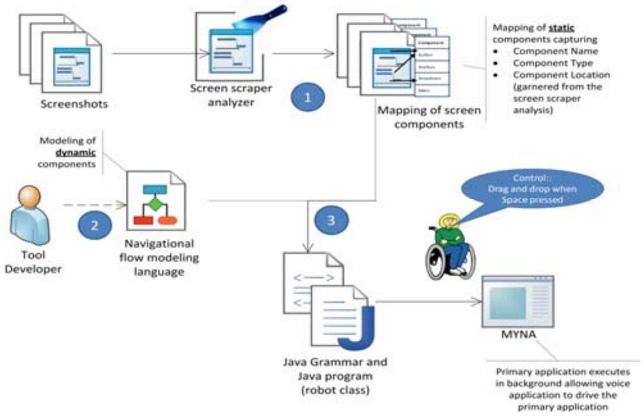
1. User says a voice command.
2. The input command is identified by the speech recognizer and checked against the grammar file.
3. If the command is present in the grammar file, an appropriate action is invoked in the Command Executor.
4. The Command Executor obtains the current mappings of the component.
5. If necessary, the Command Executor requests the Model to change its current state
6. The Command executor calls into the Java Robot to perform the corresponding mouse/keyboard action.

Current and Future Work

Phase II: Expand the existing implementation of Myna to create a 100% voice-controlled application (scroll-bar navigation, complete grammar, grammar customization).

Phase III: Evaluate Myna with a small group of peers to ensure the application is usable in order to minimize frustration from our target audience, UCP of Birmingham. Then, we will perform a formal user-study with UCP of Birmingham.

Phase IV: Using model driven engineering, we will create a model to simplify the process of incorporating voice controls into applications built with the WIMP metaphor as the core such as Lego Mindstorms Labview, App Inventor, and Alice. The below diagram is the strategy for how we will approach the generalization of our previ-


Screenshots
Screen scraper analyzer
1
Mapping of screen components
Mapping of static components capturing
• Component Name
• Component Type
• Component Location (garnered from the screen scraper analysis)
Modeling of dynamic components
Tool Developer
2
Navigational flow modeling language
3
Control:: Drag and drop when Space pressed
Java Grammar and Java program (robot class)
MYNA
Primary application executes in background allowing voice application to drive the primary application

UCP United Cerebral Palsy of Greater Birmingham — Life without limits for people with disabilities
NSF
Google research