

SCROLL OR SPEAK: A PILOT STUDY ON USER PREFERENCE BETWEEN DIRECT MANIPULATION AND VOICE INPUT WHEN ACCESSING LARGE AMOUNT OF DATA ON A PORTABLE INFORMATION APPLIANCE

Alfred Lui
School of Computer Science, Telecommunications and Information Systems
DePaul University
243 S. Wabash Avenue, Chicago IL 60604
USA

ABSTRACT

This pilot study examines user preference between direct manipulation and voice input when dealing with a large amount of data using an information appliance with a limited display. By understanding user behavior and reasons behind their preference, this study aims to establish guidelines for user interface designers wishing to enhance usability of information appliances. Based on evaluations of qualitative and quantitative data from 6 participants, voice input was found to be the input method of choice. Participants who preferred voice cited increased speed, sense of novelty and possibility of hands-free operations. Three guidelines were presented to designers working with direct manipulation and voice input in information appliances dealing with large amount of information in a limited display. While time and resource restraints created limitations, the results of this study showed promise for future research.

KEY WORDS

Information appliance, direct manipulation, voice recognition.

1 INTRODUCTION

As the means of accessing information migrate from specificity of desktop computers toward ubiquity of information appliances (IA's), integration of speech recognition has begun to gain attention in user interface designs. The differences (diversity) and changes within (variability) use environments in which IA's are operated challenge the robustness of traditional pointer-and-keyboard input methods. With the advancements in speech recognition and microelectronics technologies, speech driven interfaces are slowly making their way into portable and pervasive devices [1]. The nature of portability means that user interface of the computing devices typically have a small display and minimal number of controls. That configuration makes speech input a very attractive option, since it can occupy as little space as a microphone. Effectively incorporating speech

recognition into a visual user interface, however, takes interface designers into previously un-chartered territories.



Figure 1: User Interface of Test Application

Based on this premise, the following study explores users' decisions between direct manipulation and voice input when operating with a large amount of data. Limited display area and controls of portable devices force user interface designers to simplify data presentation, a challenge when the amount of data is larger than the display capacity of the device. While conventional wisdoms in information architecture adequately guide designers in creating visual clarity, they offer little help in promoting usability via the audio channel. The traditional

multi-layered information hierarchy actually *obstructs* voice interactions because it hides information from users and creates artificial constraints on the voice commands and user's interpretation of the feedback. Furthermore, interface designers wishing to maintain consistency between the visual and audio channels face added complexity in their decision-making process. If designers know which input method users prefer, their effort can be prioritized for the interactions required for the method.

The goal of this study is to determine user preference between direct manipulation and voice input when dealing with large amount of data, and to present guidelines for interface designers of information appliances. Previous researches on multi-model interfaces focused on developing underlying technologies (e.g. multimodal input methods, virtual reality and pure voice user interface) instead of exploring ways to apply existing technologies into usable designs [1, 2, 3]. Their findings are system-centric and solution specific [1, 2, 5]; interface designers need to digest large amount of research results and interpret them from a user-centered perspective in order to distill useful information that they can apply to their designs.

2 METHOD

2.1 Experiment Design

Six participants who are familiar with common graphical user interface controls and fluent in English were asked to perform three tasks using an information appliance

implemented on a desktop computer. All three tasks required them to navigate with the user interface and locate information from a set that is larger than the display capacity of the interface. They were asked to perform the first 2 tasks (the prescribed tasks) using only their voice or a pointing device similar to the TrackPoint™ from IBM, simulating a stylus. In the third task (the self-selected task), they were asked to relate their experience from using the 2 input methods with real-life application of the information appliance, and choose an input method that is better suited for the final task. Every task requires the participant to query against a new set of information in the test application. The prescribed tasks were also counter-balanced to avoid data context and learning effects.

For each participant, task completion time, choice of input method for self-selected task, as well as reasons for their choices were collected for analysis.

2.2 Test Application Design

The test application is a mocked-up portable information appliance user interface (Figure 1). The test application is intended to help salespeople in wholesale warehouses remotely query inventory and pricing information for customers on the shop floor. Information stored in the appliance is a fictitious warehouse store catalog with more than 120 items, and 3 inventory reports with 16-18 lines on each (Figure 2). It was developed using Visual Basic and Microsoft Speech API (Application Programming Interface) 5.1 and runs on a laptop

<u>Departments</u>	<u>Items</u>	<u>Brands</u>	<u>Reports</u>									
	Appliances (30) Air Conditioner Blender ...	Appliances (10) Basic Essentials Corning Ware ...	Appliances (18) <table border="1"> <thead> <tr> <th>Model</th> <th>InvQty</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>4SCD-39</td> <td>30</td> <td>\$30.60</td> </tr> <tr> <td>...</td> <td></td> <td></td> </tr> </tbody> </table>	Model	InvQty	Price	4SCD-39	30	\$30.60	...		
Model	InvQty	Price										
4SCD-39	30	\$30.60										
...												
Department (3) Appliances Electronics Tools	Electronics (30) Amplifier Cassette Deck ...	Electronics (10) Bose Canon HP ...	Electronics (16) <table border="1"> <thead> <tr> <th>Model</th> <th>InvQty</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>AJ2202</td> <td>901</td> <td>\$239.13</td> </tr> <tr> <td>...</td> <td></td> <td></td> </tr> </tbody> </table>	Model	InvQty	Price	AJ2202	901	\$239.13	...		
Model	InvQty	Price										
AJ2202	901	\$239.13										
...												
	Tools (30) Bench Vise Circular Saw Compressor ...	Tools (5) Craftsman Delta Magnum ...	Tools (16) <table border="1"> <thead> <tr> <th>Model</th> <th>InvQty</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>CCD-A203</td> <td>42</td> <td>\$29.50</td> </tr> <tr> <td>...</td> <td></td> <td></td> </tr> </tbody> </table>	Model	InvQty	Price	CCD-A203	42	\$29.50	...		
Model	InvQty	Price										
CCD-A203	42	\$29.50										
...												

Figure 2: Information structure in the catalog built into the test application. Parenthesized numbers are the number of selections in each category.

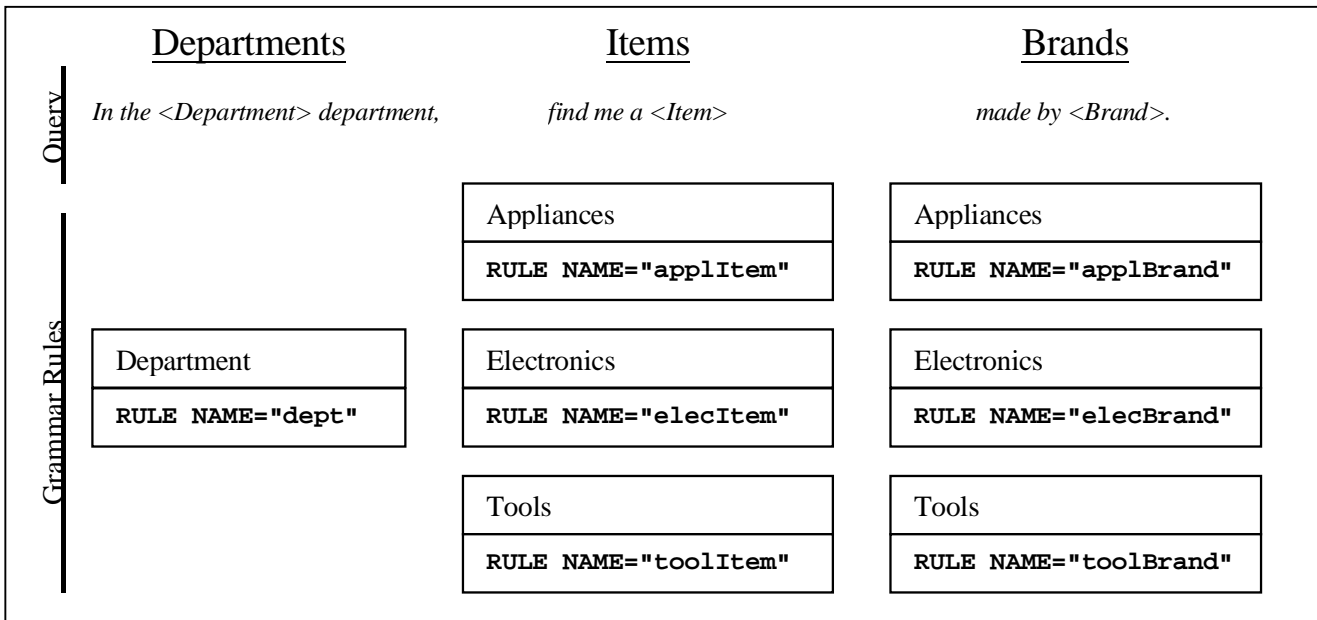


Figure 3: Seven rules related to query in relationship to information structure in the test catalog.

computer connect to a microphone headset. The user interface takes the appearance of a Palm handheld device, with a display area that can fit 18 lines of 11-point text.

At the top of the user interface is a query string:

“In the <Department> department, find me a <Item> made by <Brand>.”

Each bracketed word in the query string represents a drill-down level in the catalog (Figure 2). Below the query string is a list-box, which displays selections available for each bracketed word. To perform a query, the user makes a selection for each bracketed word in the query string. After each selection is made, the list-box is updated with the selection available for the next bracketed word. The result of the query is an inventory report displayed in the list-box.

Below the list-box is a status line that displays system messages. Next to the status string is a ‘Start Over’ button that returns the user to the beginning of a query. Two audio cues were used to help user distinguish between informational and error messages.

The test application has 2 input modes, which are controlled by a toggle-button labeled ‘Speech’ at the bottom of the user interface. In direct manipulation mode (when the toggle button is not depressed), the user controls the application using the pointing device of the laptop computer, as if the user is using a stylus on a Palm OS device. In voice-recognition mode, the user controls the application by speaking to the headset. Interactions in both operating modes were designed to match each other as closely as possible. All functions required to complete

the tasks were available in both modes and made evident to the participants prior to the experiments.

The test application has 8 grammatical rules, which specify to the Microsoft voice recognition engine what users can say. One rule included commands that directly controlled the application, and the remaining 7 represent variations of the query string (Figure 3). The rule of application commands is always active; the other 7 is turned on and off depending on user’s selection and the section of the query the user recites.

The list box can only display 10 lines at a time. The tasks were designed such that, regardless of the input method, the participant has to scroll at least once in order to complete each task.

3 RESULTS

3.1 General Observations

All the participants were able to complete their task regardless of the input method. Some errors were observed during the prescribed tasks, but all the participants who experienced problems were able to detect and recover from their errors. One type of error stemmed from participants’ attempts to recite the entire query string at once without waiting for the list-box to refresh, causing sections of their query to be rejected by the application. The second type of error was caused by the voice engine mistakenly selecting phonetically similar items from the list-box.

During the prescribed tasks, close to half of the participants attempted to use both input methods to

facilitate the lookup from the list-box. That behavior was not observed during the self-selected task.

3.2 Quantitative and Qualitative Results

Table 1 summarizes quantitative data collected from the experiments. Five out of the 6 participants chose voice input over direct manipulation in their self-selected tasks. In the qualitative response of those 5 users, 4 them reported that speed was a reason of their choice. Secondary to speed, novelty (3 of the 5) and ability to free up their hands (2 of the 5) also influenced their decisions. Three of them indicated that voice input took sometime for them to learn, but with practice, they found it to be easier to use than direct manipulation. Only 1 out of the 5 participants reported that accuracy was a reason for choosing voice input.

For the 1 participant who chose to use the point device in the self-selected task, speed, accuracy, familiarity with pointing device and doubt on the performance of voice recognition in a high-noise environment were the reported reasons.

Table 2: Mean and standard deviation of task completion times (in seconds).

	Prescribed		Self-selected	
	Pointing Device	Voice	Pointing Device	Voice
Mean	38	44	18	22
S.D.	10	26	N/A [†]	10

[†]Not applicable because only 1 data-point is available.

Regardless of the input method, there was an approximate 100% improvement in task completion time between prescribed and self-selected task (Table 2). That improvement can be attributed to user’s increased familiarity and confidence with the user interface.

Interestingly, although participants who chose voice for it’s speed, they actually performed slower than the 1 who chose pointing device by 4 seconds (or 22% slower). It is possible that other factors such as novelty and hands-free operations made up for the longer task completion time.

Participant	Guided			Self-selected		
	Pointing Device (t_{pd})	Voice (t_v)	$t_{pd} - t_v$	Input Method	Time	Reason(s)
One	30	20	10.00	Voice	15	Speed, Novelty
Two	40	55	-15.00	Voice	30	Speed, Novelty, Hands-free Operation
Three	35	90	-55.00	Voice	17	Speed, Novelty, Hands-free Operation
Four	55	26	29.00	Voice	13	Speed, Novelty
Five	28	30	-2.00	Point Device	18	Speed, Accuracy, Familiarity
Six	42	40	2.00	Voice	35	Speed, Accuracy, Novelty

Table 1: Task completion time (in seconds) and participant’s choice on the self-selected task.

4 DISCUSSION

4.1 Direct Manipulation vs. Voice Input

Results from this pilot study suggests that with limited display commonly seen on portable information appliances, voice input is the input method of choice. Users’ familiarity with the pointing device operations in graphical user interfaces was super-ceded by quicker time-to-task, sense of novelty, and hands-free operation offered by voice input. Although in the experiment, participants who chose voice actually performed slower (which can vary depending on the interaction) other benefits are attractive enough to sway user to prefer voice input.

These findings are significant to user interface designers of information appliances because it suggests that with limited display devices, voice input can potentially be the only input method. Elimination of other components needed for direct manipulation, such as touch sensitive screens and keyboard, may not only simplify interactions but also reduces the physical size of the appliance. With user interfaces that have a limited display and require both direct manipulation and voice input methods, designers can focus their efforts by putting their priorities in optimizing voice interactions.

4.2 Speed of Voice Input

In this study, users preferred voice input for its speed, this means that user interface designers should optimize user interactions toward shorter task completion time. The challenge in having both direct manipulation and voice query lies in effectively amalgamating visual information architecture and query options onto a limited display, such that users who want to use direct query knows the query structure and available options, while users who want to explore the application (using either voice or direct manipulations) can still do so using the graphical user interface. In the test application, the query string is prominently displayed at the top. The list-box below the query refreshes after each user selection and signals the user with all the available options at each successful input. Both the 100% task completion rate and 50% improvement in task completion time suggest that the

strategy is successful and is an option that user interface designers can consider.

In the case where permanently displaying the query string is not possible, user interface designers should design voice queries around user's mental model and domain knowledge. Directly translating traditional GUI elements (e.g. menus, icons and buttons) into grammar for the voice recognition engine and expecting that users are able to derive the proper voice command from separated visual elements on the display is known to be error-prone and frustrating to users [5].

4.3 Weakness of Voice in Providing Feedback

Designers should avoid relying on speech synthesis to provide system feedback unless the device has an eyes-free application (in telematics systems, for example). Visual channel is much more efficient in communicating system status when the user can read the device. Speech on the other hand, is ambiguous [6], serial and short-term memory intensive [4]. Users cannot easily skip back or scan forward in verbal messages as they can on the displayed messages. Using synthesized speech as feedback also poses technical challenge. If the user wants to issue a command in the middle of a synthesized system message, the device must be able to interrupt or ignore the message. Otherwise, the system message can create interference in the user's input [5]. Simple and short audio cues coupled with textual message displayed at a fixed location worked well on the test application. Users knew where to look in response to an audio cue; they can read and choose to respond or ignore the message at their own speed.

4.4 Multi-Modal Input

Although participations were asked to use only 1 input method (either given or chosen) in each task, almost half of them tried to switch back and forth between methods to facilitate searching and scrolling in the list box during their prescribed tasks. It was because that type of behavior was not the focus of this study, insufficient data is available to determine the reason of that behavior. Nevertheless, user interface designers should accommodate for multi-modal input method in their designs, especially for novice users who are not familiar with capabilities of the device.

4.5 Designing for Voice

To achieve successful speaker independent voice recognition (i.e. without requiring user to training the engine), the voice recognition engine in the Microsoft Speech API relies on a set of rules provided by the application. Those rules, or context-free grammar, specify the structure and variation of phrases the voice recognition can expect to receive from the user. In other

words, the grammatical rules define what the users should say in order for their input to be recognized by the engine. An effective implementation of voice recognition requires defining and tuning the grammar such as phrases are specific and different enough to be recognized by the computer, and at the same time they are flexible and natural enough that users do not feel restricted on what they can say. That fine balance can be achieved by observing linguistic features adapted by user constituents and tailoring voice commands around the their language. Another common technique to increase recognition of voice commands is to create phonetic differences by embedding multi-syllabic words.

In the test application, the query was separated into sections and each section was assigned phonetically different words. That strategy has help minimizing errors and rejections in the voice recognition engine. Errors caused by participant's recital of the entire query without pausing after each section were easily detected and later avoided by the participants. Errors caused by the voice engine mistakenly picking phonetically similar selections within the list-box were more disruptive. For example, one of the participants said 'Tile Saw' in the self-selected task but 'File' appeared in the Item box. It was because the voice recognition accepted the voice input, no feedback about the error was provided to the participant. The participant could not detect the wrong selection and continued with the task until the experimenter intervened. Although the participant successfully avoided the error in the second trial, completion time of that time was dramatically increased. Organizing pieces of information in such a visually clear way while separating phonetically similar ones were not issues explored in the design of this test application. However, it is a realistic and fundamental issue that user interface designers must consider in their work.

5 CONCLUSION AND FUTURE WORK

Based on the analysis of the data collected from the experiment, this study offers the following design guidelines to user interface designers of information appliances that support voice and direct manipulation input methods, and with limited display areas that need to accommodate larger amount of data:

- Voice was the input method of choice, with increased speed, sense of novelty and possibility of hands-free operations as noted reasons. Thus, voice input should be promoted in user interface design.

- Design interactions that support multi-modal input. Users who are not familiar with the user interface and/or information provided by the information appliances are likely to use both direct manipulation and voice input.
- Organize on-screen information to create visual clarity and to separate (or re-categorize) phonetically similar information. Simply sorting items can result in clusters of choices that sound similar in voice commands, reducing accuracy of voice recognition.

At the time of this study, there are few literatures that offer clear guidance on designing user interfaces that have both direct manipulation and voice input capabilities. Most of the previous literature focused on areas such as graphical user interfaces, speech interfaces and sound design. This study was intended to provide guidelines for designers to make noticeable usability improvements to their work. While it is believed that such intention was met with the three guidelines provided above, many questions remain for future examination. Strategies to effectively support multi-modal input and how to organize information for both visual clarity and promotional of phonetic differentiation in voice commands are two readily available opportunities for future research. Answers to these questions can dramatically improve usability of future information appliances.

Although many considerations were put forth on the design of the experiment and analysis of the results, time constraints have created limitations on this study. The test application was meant to be conducted on a portable information appliance, but resource and time constraints did not permit it to be implemented on a portable device. There are documented differences between mouse and stylus on user performance [7]. Using a point-device may have affected participants' perception on the speed of direct manipulation. The small sample size in this pilot study is another limitation. Study data showed a pattern of user preference for voice but sample size was too small for statistical analysis. Only 1 participant reported preference in direct manipulation. This pattern of voice preference over direct input can be further explored in a larger study.

REFERENCES

[1] S. Oviatt & P. Cohen, Multimodal Interfaces that Process What Comes Naturally, *Communications of the ACM*, 43(3), 2000.

[2] C. Schmandt & E. A. Hulstén, The intelligent voice-interactive interface: *Proceedings of the SIGCHI*

conference on Human factors in computing systems, 1982.

[3] P. R. Cohen, The role of natural language in a multimodal interface: *Proceedings of the 5th annual ACM symposium on User interface software and technology*, 1992.

[4] D. Hindus, B. Arons, L. Stifelman, B. Gaver, E. Mynatt & M. Back, Designing auditory interactions for PDAs: *Proceedings of the 8th annual ACM symposium on User interface and software technology*, 1995.

[5] N. Yankelovich, G.A. Levow & M. Marx, Designing SpeechActs: Issues in Speech User Interfaces: *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995.

[6] A. Smith , J. Dunaway , P. Demasco & D. Peischl, Multimodal input for computer access and augmentative communication: *Proceedings of the second annual ACM conference on Assistive technologies*, 1996.

[7] Y. Guiard, M. Beaudouin-Lafon, D. Mottet, Navigation as Multiscale Pointing: Extending Fitts' Model to Very High Precision Tasks, *Proceedings of the SIGCHI conference on Human factors in computing systems*, Pittsburgh, PA, 1999.