View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings

Americas Conference on Information Systems (AMCIS)

2005

User Error Handling Strategies on a Non Visual Multimodal Interface: Preliminary Results from an Exploratory Study

Xiaoyu Chen
New Jersey Institute of Technology, xc3@njit.edu

Marilyn Tremaine
New Jersey Institute of Technology, tremaine@njit.edu

Follow this and additional works at: http://aisel.aisnet.org/amcis2005

Recommended Citation

Chen, Xiaoyu and Tremaine, Marilyn, "User Error Handling Strategies on a Non Visual Multimodal Interface: Preliminary Results from an Exploratory Study" (2005). *AMCIS 2005 Proceedings*. 299.
http://aisel.aisnet.org/amcis2005/299

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

User Error Handling Strategies on a Non-Visual Multimodal Interface: Preliminary Results from an Exploratory Study

Xiaoyu Chen
Information Systems Department
New Jersey Institute of Technology
xc3@njit.edu

Marilyn Tremaine
Information Systems Department
New Jersey Institute of Technology
tremaine@njit.edu

ABSTRACT

The present study addresses two questions: On a non-visual multimodal interface for textual information browsing, (1) how prevalent is input modality switching as an error handling strategy, and (2) how much does an input modality need to fail before input modality switching occurs. The results indicate that although switching input modalities to correct errors is an expected practice on multimodal GUIs, it is not the prevalent strategy for non-visual multimodal interfaces. We believe that users are more likely to diversify their error handling strategies within a modality, if different strategies are possible, but we have not found conclusive evidence for this belief. However, our analysis suggests that the failure to switch modalities when errors occur may, in part, be due to the prevalence of alternative error handling strategies in a particular input modality, that is, the user prefers to stay in the same modality rather than assume the cognitive load of a switch.

Keywords

Multimodal input, interface design for blind and visually impaired users, speech and tactile input, error handling strategies.

INTRODUCTION

Since the concept of universal accessibility has emerged in the fields of human-computer interaction and interface design, a variety of non-visual information systems have been developed for blind and visually impaired users to access information. Researchers' and developers' efforts have resulted in systems presenting outputs to users in aural or tactile forms, or a combination of the two. In contrast to abundant implementations of multimodal output, multimodal input has not been broadly adopted in non-visual interfaces. Keyboards and keypads still dominant the input for these systems.

We believe that multimodal input has the potential to benefit blind and visually impaired users because "well-designed multimodal systems should be able to integrate complementary modalities to yield a highly synergistic blend in which the strengths of each mode are capitalized upon and used to overcome weaknesses in the other" (Oviatt, 1999b). Empirical research is needed to provide knowledge in how to best integrate different input modalities into a complete non-visual system. One component of the needed knowledge is the understanding of user error handling strategies in non-visual multimodal systems.

Error Handling Using Multimodal Input

One of the major benefits of multimodal systems is their robustness to input recognition errors, especially speech. Various studies on multimodal interfaces designed for GUIs and sighted users have reported robust user error handling strategies.

After learning to use a multimodal interface, users select the input mode that they think is less error prone for specific tasks (Oviatt and Cohen 2000). For example, when inputting a foreign surname, users are more likely to write the name than speak it (Oviatt and Olsen, 1994). It has been found that although users originally preferred one input modality, they also used other modalities to avoid the ineffectiveness of using their preferred modality on tasks where usage was awkward (Suhm, Myers, and Waibel, 2001).

From a usability standpoint, multiple input modalities offer an error-handling advantage so that when one input mode fails the user can switch to the second one. One type of failure is a system recognition error in which the input data is misinterpreted by the system. This occurs most frequently with speech input when a system recognizes a different command than what was intended by the user. When a recognition error occurs, users tend to switch input modality, which facilitates error recovery (Karat, Halverson, Horn and Karat, 1999; Oviatt 1999b; Sears, Feng, Oseitutu and Karat, 2003). It has been reported that the likelihood that users will switch input modalities following a system recognition error is 3 times more than normal Oviatt, Laniran, Bernard, and Levow, 1998).

In studies on speech recognition systems researchers discovered an initial re-dictation strategy followed by an input modality switch when re-dictation failed (Oviatt and VenGent, 1996; Halverson, Horn, Karat and Karat, 1999).

To offer full error-handling advantages multimodal system design should follow two design approaches. First, different input modalities need to provide parallel or duplicate functionality to allow the user to freely switch between input methods at any time. Second, different input modalities should be used to disambiguate each other so that recognition errors from unimodal recognition can be mitigated (Oviatt 1999a).

In a study on a semi-simulated electronic map system that accepts speech input and pen inputs, Oviatt (1999b) reported that one out of eight commands processed by the multimodal system produced the correct response because of mutual disambiguation between the speech and pen inputs. Oviatt (1996) also reported that integrated speech and pen input resulted in 36 percent fewer task errors, 23 percent fewer spoken words, and 10 percent faster completion time compared to speech input only.

In a study comparing various unimodal and multimodal error correction strategies for speech user interfaces, Suhm and his colleagues confirmed that cross-modal repair speeded up the correction of speech recognition errors on speech user interfaces compared to unimodal error correction by respeaking, and that cross-modal repair was more accurate than unimodal repair (Suhm et al. 2001).

Goal of the Study

Although positive results regarding error handling have been reported as above, those results were obtained on multimodal interfaces designed for GUIs and sighted users in the application domains of interactive map systems and speech error correction on speech interfaces. It is unknown yet to what extent these results can be applied to non-visual multimodal interfaces, and to information browsing domains that do not have a physical domain context.

The present study aims to extend our understandings of users' error handling strategies when using non-visual multimodal input to perform textual information browsing tasks. The study aims to provide insights to the following questions:

On a non-visual multimodal interface for textual information browsing,

- How prevalent is input modality switching following an input failure?
- How resistant is a user to switching input modalities when the input modality is failing?

In the present study, apart from analyzing users' error handling strategies, system recognition errors are classified, and recognition rates are measured for speech and tactile inputs, the two input modalities integrated in our non-visual multimodal system.

The present study is exploratory. It investigates the behavior patterns of sighted users on AudioBrowser, a non-visual multimodal information-browsing system. Although the study was conducted with sighted subjects, the results in this study can serve as the baseline for comparison and thus, for understanding the behavior of blind and visually impaired users for whom the interface is being designed. Since most system developers are sighted, understanding this difference between sighted users and blind and visually impaired users has important practical implications. The present empirical study is a preliminary step toward learning how to combine different input modes into a strategically integrated whole non-visual system.

The rest of this paper is organized into five sections: the SYSTEM DESCRIPTION section describes the non-visual multimodal interface implemented for textual information browsing and used for the study. The SUBJECTS, TASKS, AND PROCEDURE section introduces the study methods. The RESULTS and DISCUSSION sections present the experiment results and then discuss the implications of these results. The paper then concludes with a FURTHER RESEARCH section.

SYSTEM DESCRIPTION

AudioBrowser is a non-visual information browser that organizes information, e.g. news articles, into a hierarchy and reads the information for users. Users can navigate the hierarchy and control the manner in which the contents are read through a combination of speech and tactile input commands. User controls include browsing the information hierarchy, reading the content of an item in the hierarchy, checking one's location in the information space, pausing and resuming reading, having the system read one word, sentence, paragraph or complete article at a time, spelling out words, repeating, skipping and proceeding to the next or previous sentence or paragraph of text, and adjusting settings for auditory output, such as reading speed, volume, reading voice and pitch, and the volume of non-speech sounds.

Tactile input inherits the design of SoundNews (Williams and Tremaine, 2001). It consists of pointing gestures on the sensing area of the touchpad, and clicks on the buttons beside the sensing area. Speech input parallels tactile input. It is received via a microphone and processed by the Microsoft™ Speech Recognition Engine. All system outputs are speech or non-speech audio. Speech outputs are standard American English synthesized by the Microsoft™ Text-to-Speech Engine. Non-speech outputs such as clicks and swishes are prerecorded audio clips.

Tactile input is performed on a Synaptics™ touchpad (figure 1). The API provided by Synaptics™ allows developers to program functions for the sensing area and the buttons. The sensing area of the touchpad is divided into three tracks. The borders of the tracks are marked using paper clips that are scotch-taped to the pad. These tracks provide physical cues that guide the visually impaired users to the desired track. The two buttons at the two sides of the sensing area are used to execute commands. The two buttons under the sensing area are currently not used.

Figure 1. Programmed Synaptics Touchpad

The tracks are dynamically divided into small virtual segments, as illustrated in figure 2. Each segment is mapped to an information item or an operation command. Once a segment is touched, the system speaks aloud the corresponding item on it.

The first track is dedicated to browsing stored information. Gliding a finger across the track explores a single level of information items. Pressing the left or the right button traverses up or down the information hierarchy. (Figure 2)

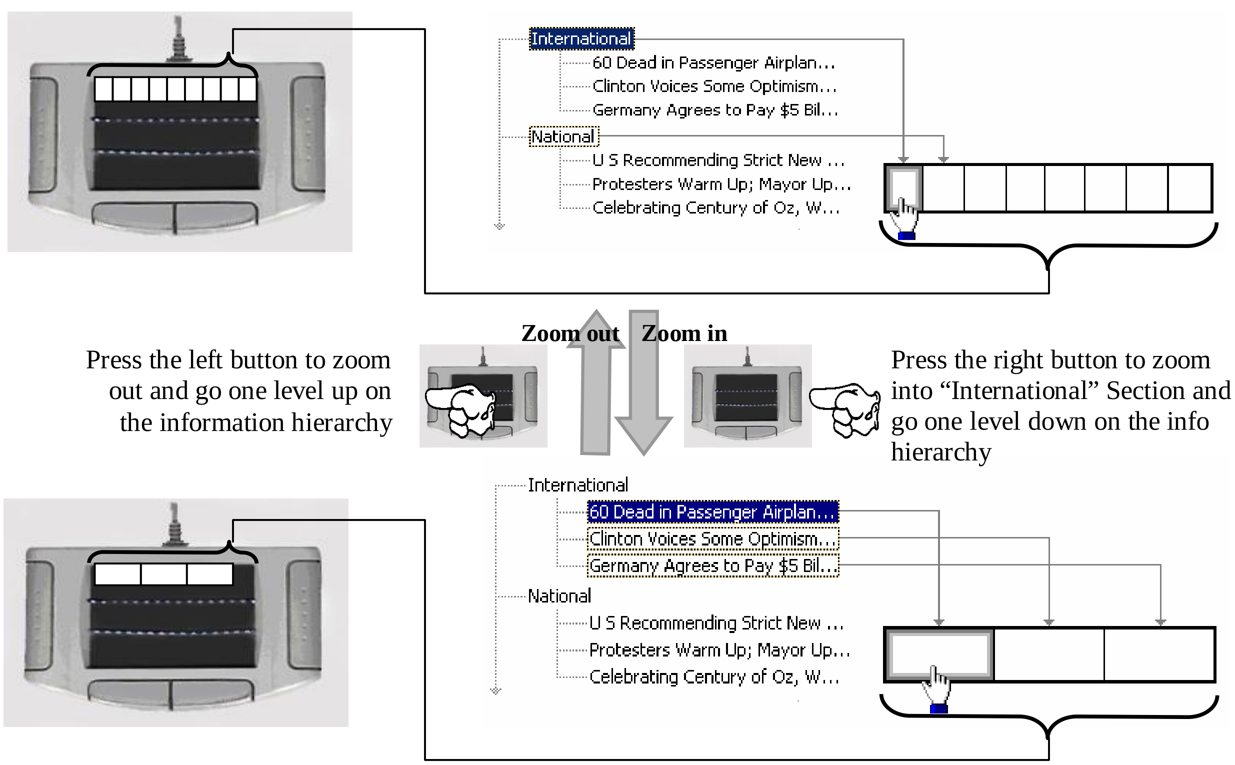

Figure 2. Browsing Hierarchical Information Using the Touchpad

The second and the third tracks are mapped to operation commands. Users explore these tracks to find a desired command and then press one of the buttons to execute the command. Button presses are mode sensitive, i.e. what function a button press performs depends on what item is selected on the tracks. For example, when the command "read by sentence" is selected, a right button press causes the system to enter "read by sentence" mode and to read the next sentence. A left button press reads the previous sentence. When "change reading volume" is selected, a right button press increases the reading volume and a left button press decreases the volume. The modal design was chosen as a solution to deploy needed functions on a limited operation space (The Synaptics™ touchpad has only two working buttons.) This design choice was made to also not lose the intuitivism and consistency of the operations, as pressing the right button always results in the system reading the "next" item or increasing a selected value, and pressing the left button always results in reading the "previous" item or decreasing a selected value.

Speech commands are executed when users hold down a "push-to-talk" button and speak the command into the microphone. Only one speech command will be executed for each hold of the button. We used this solution because the system is also constantly speaking to the user and the speech interferes with the input commands. (Note that this problem can also be fixed by more complicated microphone setups.) Rather than using complicated commands such as "Find telephone number of John Smith," we created simple and short speech commands that paralleled the tactile commands on the system menus. This allowed any input operation to be given by either speech input or tactile input. An input operator is the smallest user input unit. A typical user task usually consists of a series of input operators. Thus, a user task can be performed using either multimodal input or unimodal input.

There are three reasons for creating parallel speech and touchpad commands at the operator level. First, the user can decide to finish a user task using the easiest and most efficient input modality combinations. Second, at any time when one input modality fails, the user can switch to the other modality to recover from the failure. Third, speech command that is forgotten can always be found on the touchpad paralleling a similar advantage that is found in graphical user interfaces.

SUBJECTS, TASKS, AND PROCEDURE

Fifteen subjects participated in the study. During the study one subject did not speak fluent enough English to use our speech recognizer. Hence, the data for this subject was excluded from our analysis. The remaining fourteen subjects all spoke fluent English. At the time of the study, four subjects held a masters degree or were in graduate school, and ten were undergraduates. All subjects had taken or were taking computer-related courses.

The subjects were sighted but were not provided any visual interface during the study. The only feedback from the system was auditory. The study was conducted in a closed lab environment. The complete study took place in three consecutive days, with each subject participating individually. The total participation time of each subject was about six hours.

In the first two days, the subjects participated in two tutorials for using the speech input and the tactile input. Preceding the speech input tutorial, each subject trained the speech recognition engine for thirty minutes. To counterbalance the training order effects, half of the subjects received the speech input tutorial on the first day and the tactile input on the second day. The other half received the tutorials in the reverse order. The subjects were assigned to the two training orders randomly. During the tutorials, subjects read the text tutorials and tried the system functions as instructed by the tutorial document. Following each tutorial session, a practice session was held in which the subjects performed a list of tasks that covered all the system functions using the input method taught in the preceding tutorial session.

On the second day, after the subjects had finished both speech and tactile input tutorials, they engaged in a third practice session in which they were asked to freely mix the two input methods. This session was designed to allow subjects to develop efficient ways for combining both input modes for different user tasks.

On the third day, the subjects engaged in a warm-up session using both input modes. They then participated in the experiment session. They were given a new list of tasks that covered the complete system functions and represented the typical use of the AudioBrowser system. The subjects could use any combination of input modes to perform the tasks.

Each session was videotaped. The videos were then analyzed to determine how users mixed the modalities given the tasks and their prior training.

RESULTS

A total number of 1642 input operators were performed by the 14 subjects during the experiment sessions, among which 635 or 38.67% were speech operators, and 1007 or 61.33% were tactile operators. The total number of input operators per subject averaged 117.3 and ranged from 73 to 200, with a standard deviation of 29.4. The number of speech input operators per subject averaged 45.4 and ranged from 5 to 81, with a standard deviation of 24.9. The number of tactile operators averaged 71.9 and ranged from 27 to 129, with a standard deviation of 30.8. Mode choices by individual subjects were very diverse, but the types of input errors were common, and some patterns in error handling behaviors were derived.

Input Error Analysis

Errors occurred in both speech input and tactile input. The success rate of speech input during the experiment was 73.7%, and that of tactile input was 95.6%. We defined as an error all those operations that were unsuccessful, or partially successful (i.e., succeeded with problems). There were 167 unsuccessful speech operators and 44 unsuccessful tactile operators in total.

Success with problems: The partially successful input operators resulted in successful system interpretations and responses that were not understood by the user primarily because of the unnaturalness and the phrasing of the machine speech generated by the text-to-speech engine used.

Complete failure: The completely failed input operators represented situations where the system gave an explicit error message, responded wrongly because of recognition errors, or provided no feedback at all because it the system did not recognize that a command was being given. The failure types in speech input and tactile input are not completely the same.

In speech input, six major types of failures were identified. (1) *Speech output interference*: When the user gave a speech command, the speech from the system interfered with the speech input. A typical scenario of this type of failure occurred when the user said "Pause" while the system was reading text. A word from the system's speech was used as the command and thus, the command was recognized incorrectly. (2) *Speech recognition errors*: The speech recognition software failed to recognize a speech utterance of the user. (3) *No-response problem*: The system did not respond to a speech input with any feedback until the user repeated the command or gave another input command. The likely cause of this problem is system delay associated with a background task consuming the computing resources needed by the speech synthesis application. (4) *Environment noise interference*: When the push-to-talk-button was pressed, the microphone picked up environment noise, such as unrelated speech by the subject or noise coming from the next room. This changed the auditory input causing the speech recognition engine to recognize a different command then that spoken. (5) *Incorrect user mental model*: The user understood the system differently from how the system actually worked. The user therefore gave commands that were wrong for the user's goal accomplishment. (6) *Other failures*: Occasionally the failure was not any of the above types.

In tactile input, the failures were (1) caused by an incorrect user mental model, and (2) input mode errors. Mode errors were caused by the mode-sensitive design of the touchpad buttons, i.e., the function that a button press executes depends on the last segment that the user has touched on the touchpad tracks. Users often forgot what system mode had been set last on the touchpad causing their button presses to lead to unexpected system actions.

Tables 1 and 2 provide an overview of successful and unsuccessful operators of speech and tactile input respectively.

Summary Items		Count	Percentage
Total No. of Speech Input Operations by All Subjects		635	100%
No. of Successful Speech Operations		468	73.70% in total no. of speech Operators
No. of Partially Successful Speech Operations		10	1.57% in total no. of speech Operators
Completely Failed Speech Operations	Total No. of Complete Failures	157	24.72% in total no. of speech Operators
	(1) Speech output interference	47	29.94% in total no. of speech failures
	(2) Speech recognition errors	4	2.55% in total no. of speech failures
	(3) No response problem	66	42.04% in total no. of speech failures
	(4) Environment noise interference	15	9.55% in total no. of speech failures
	(5) Incorrect user mental model	3	1.91% in total no. of speech failures
	(6) Other system failures	22	14.01% in total num. of speech failures

Table 1. Summary of Success and Failure in Speech Input

Summary Items		Count	Percentage
Total No. of Tactile Operators by All Subjects		1007	100%
No. of Successful Tactile Operators		963	95.63% in total no. of tactile operators
No. of Partially Successful Tactile Operators		2	0.20% in total no. of tactile operators
Completely Failed Tactile Operators	Total No. of Complete Failures	42	4.17% in total no. of tactile operators
	(1) Incorrect user mental model	27	64.29% in total no. of tactile failures
	(2) Mode errors	15	35.71% in total no. of tactile failures

Table 2. Summary of Success and Failure in Tactile Input

Error Handling Actions

Users' error handling actions were analyzed at two levels, the *operator level* and the *error correction case level.* At the operator level, the researchers recorded each input operation that immediately followed each input failure. At the error correction case level, the researchers recorded each sequence of error correction operations that took place until the error was corrected or given up.

Results at the Operator Level

In speech input, a total number of 167 unsuccessful (i.e. partially successful + complete failed) operations occurred for all fourteen subjects. Among those unsuccessful operations, 139 or 83.23% were followed immediately by a speech input that attempted to repair the error. Only 28 or 16.77% were followed immediately by a tactile input for this purpose.

A total number of 44 unsuccessful tactile operations occurred for all subjects. 37 or 84.09% unsuccessful tactile operations were followed by a tactile operation and 7 or 15.92% were followed by a speech operation to repair the error.

Situations in which an unsuccessful speech operation is followed by another speech operation are coded as **ss**; situations where an unsuccessful speech operation is followed by a tactile operation are coded as **st**; an unsuccessful tactile operation followed by another tactile operation is coded as **tt**; and an unsuccessful tactile operation followed by a speech operation is coded as **ts**. The counts of occurrences of **ss**, **st**, **tt**, and **ts** for each subject are summarized in Tables 3 and 4.

A paired t-test was conducted to compare the occurrences of **ss** and **st** by subjects. Users were found to significantly use speech to repair speech errors rather than switching to touch (t (13) = 4.279, p < 0.001). A second paired t-test was conducted to compare the occurrences of **tt** and **ts** by subjects. Users were found to use the tactile operation to repair tactile errors significantly more times than switching to speech (t (13) = 2.760, with p < 0.05).

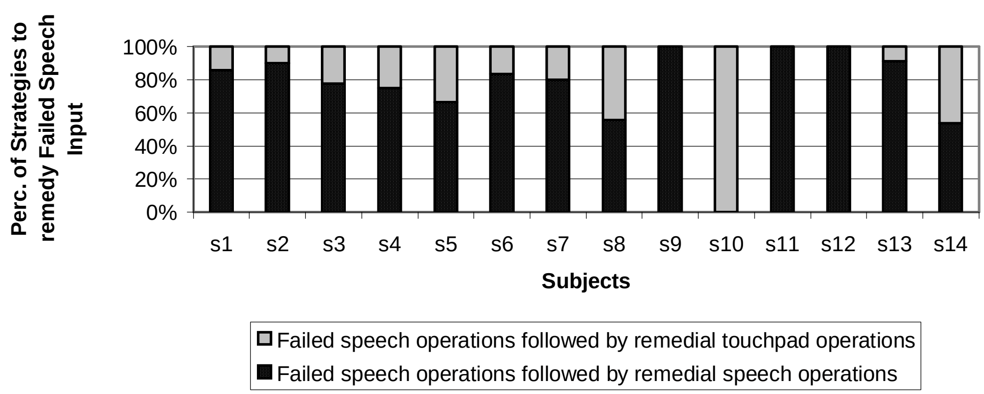

Figure 3. Use of Error Handling Strategies (in Percentages) for Failed Speech Input Operations

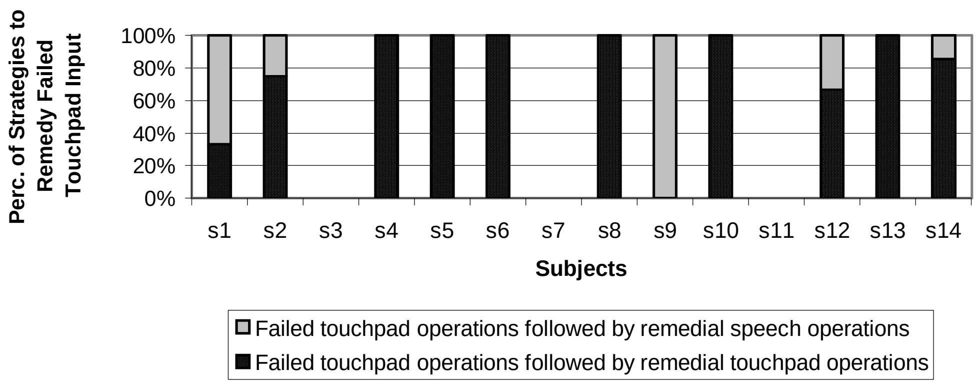

Figure 4. Use of Error Handling Strategies (in Percentages) for Failed Tactile Input Operations

The t-tests and the charts show clearly that in most cases, when an input operation failed, the subjects did not switch input modality but continued to use the same input modality for error recovery. Exceptions occurred for Subject 10 in speech error recovery (Figure 3) and Subject 9 in tactile error recovery (Figure 4). For these two subjects, only one input operation error was encountered, and the subjects switched their input modality to handle the problem.

Results at the Error Correction Case Level

At the error correction case level, the researchers recorded each sequence of error correction operations that occurred until the error was corrected or given up. Each sequence of error correction attempts forms an **error correction case**. In each error correction case (i.e. following each error), there could be more than one unsuccessful operation (i.e. unsuccessful error correction attempts), and there could be none or one successful operation (i.e. successful error correction attempt).

For all subjects, a total number of 158 errors occurred during the experiment sessions. The subjects attempted to correct 150 of them and did not try to correct 8 of them. Thus, there were 150 error correction cases. In each case, error correction operators did not always succeed at the first attempt. It took one to five operations (i.e. attempts) to correct an error. The average attempts to correct each error were 1.32. The distribution of the error correction attempts is shown in Figure 5.

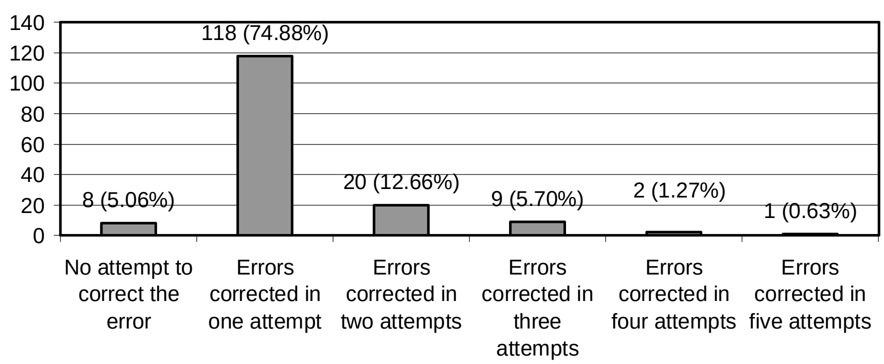

Figure 5. Counts of Cases that an Error was Corrected in One, Two, Three, Four or Five Attempts

Our results are different from our prediction and from other published research. In our non-visual system, when an error occurs in a particular input modality, users will tend to use the same input modality to correct the error. Overall, the number of error correction cases that involved input modality switching were significantly less than the number of cases in which the error was corrected without an input modality switch. Among the 150 error correction cases, 34 (i.e. 22.67%) involved input modality switches while 116 (i.e., 77.33%) did not. A paired t-test found that users formed significantly more error corrections without an input modality switch ($t(13) = -4.519$, $p < 0.001$).

To understand the above results better, a more detailed analysis was conducted. The AudioBrowser system not only allowed users to perform input operations in either touch or speech, but also often had multiple methods for performing the same task within an input modality. Thus, two error correction situations were identified. In the first situation, only one method is available to perform the task that came up with an error. In the second situation, the user can switch to another method to accomplish the same task.

The first situation occurred 62 times. In the first situation, in 19 or 30.65% of the cases users switched input mode, while in 43 or 69.35% of the cases, users did not switch input mode. A paired t-test found that users switched modalities significantly less for the single method case (t (13) = -2.604, p (one-tailed) = 0.01). Figure 6 illustrates the situation.

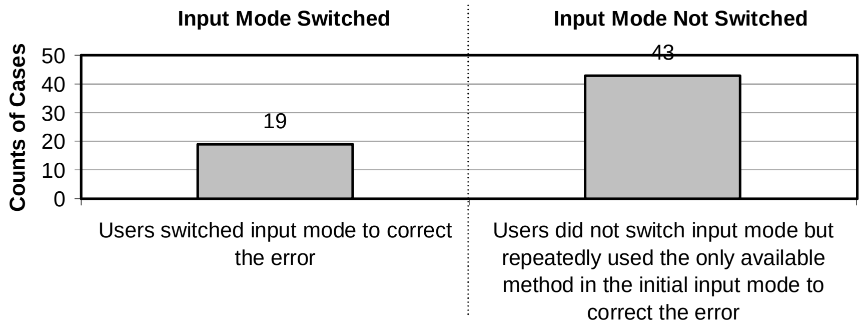

Figure 6. Types of Error Correction Strategies by Users When Only One Method was Available for Error Correction in the Initial Input Mode

The second situation occurred 88 times in total. In the second situation, when there is more than one method available for correcting the error using the current input mode, in 15 or 17.05% of the cases, users switched input modes, while in 73 or 82.95% of the cases users did not switch input mode for error correction. Input mode switches occurred significantly less than no input mode switches for error correction (t (13) = -4.833, p < 0.001). Several types of users' error correction strategies were identified from the video analysis. They are showed in Figure 7.

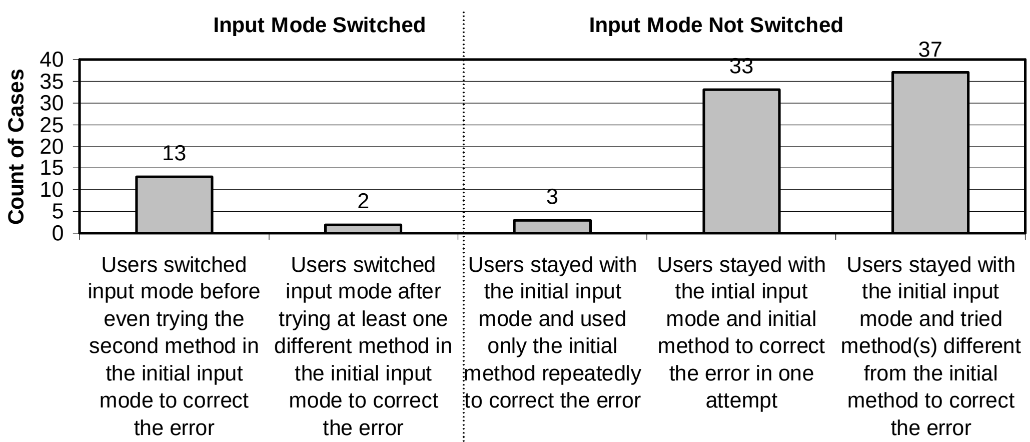

Figure 7. Types of Error Correction Strategies by Users When More than One Method was Available for Error Correction in the Initial Input Mode

We have shown that resistance to modality switching was significant for both situations. To see if the lack of multiple methods had an impact, we formed a ratio of non-switched error corrections to all error corrections for each subject and ran a paired t-test to see if these results were significantly different for the single method vs. the multiple methods situations. Our results (t (14) = 0.82, p =0.21) did not find significant differences. Since a quarter of our subjects only had one or zero errors to correct in either situation, it is likely that we were not able to find such differences with our analysis if they did exist.

The input modality was switched in a total of 34 error correction cases. In 23 cases, the input modality was switched at the first error correction attempt and the error was corrected at the first attempt. In 6 cases, the input modality was switched at the second error correction attempt and the error was corrected at the second attempt. The time points when the input modality was switched and the error was corrected are illustrated in figure 8.

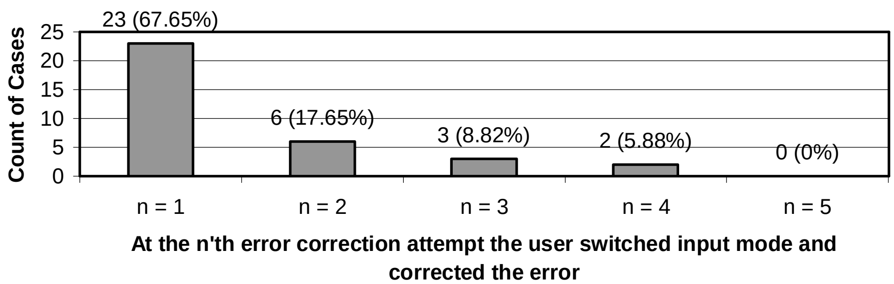

Figure 8. The Time Point at which Input Modality was Switched and Error Corrected

DISCUSSION

This exploratory study has investigated types of input errors that can occur in speech and tactile multimodal input systems. The type and number of speech input errors is representative of speech recognition applications. This study has demonstrated that to correct these errors, users use different error handling strategies. The study has provided answers to the research questions stated at the beginning of the article:

On a non-visual multimodal interface for textual information browsing,

- *How prevalent is input modality switching following an input failure?*

At the operator level, immediately following a failed input, input modality switches occurred significantly less than input modality non-switches. At the error correction level, in a sequence of error handling actions, input modality switching was still significantly less likely to happen than error handling without input modality switching.

So the answer to the question is that input modality switching in the non-visual multimodal system was not a prevalent error handling strategy. This result is different from the results obtained on multimodal GUI interfaces discussed at the beginning of this article. The reasons for this different result could be (1) that the amount of mental work involved in input modality switching has prevented switching in our system (note that the user has to search for a tactile command), (2) the lack of visual aids for our subjects meant that they could readily lose their place in the information structure if they switched modalities to correct a speech error, and (3) the availability of multiple methods for correcting an input error has encouraged the use of different methods in a single modality instead of switching modalities.

Moreover, although speech recognition had higher error rates than touchpad recognition, speech errors did not lead to higher rates of input modality switching for error correction than tactile errors. In other words, higher recognition failures do not necessarily lead to input modality switching.

- *How resistant is a user to switching input modalities when the input modality is failing?*

We believed that if there is more than one method available for error correction within an input modality, users would be more likely to stick to this input modality and switch methods instead. However, we were not able to find conclusive results for this belief. Although the percentage of non-switches favors multiple method situations, we did not have enough error data to demonstrate this conclusively.

We did show that there is a tendency to switch input modalities, the longer one input mode continues to fail during error recover. This can be seen in figures 5 and 8.

Based on the results of this study, the error handling strategy used on multimodal GUIs and non-visual multimodal interfaces are different. Input modality switching is not likely to be a major error handling strategy on non-visual multimodal interfaces. Thus, if users are to take advantage of multimodal error correction, they will need to be taught such strategies, since our study does not find them being naturally used.

What can we say about our sighted users given the results we have obtained? First, it is likely that some of the speech errors that they experienced are present because of their inexperience with listening to text to speech output. It is likely that these errors will not be as prevalent with our visually impaired users who are accustomed to this type of output. Second, it is possible that the resistance to modality switching may also be a result of inexperience with other aspects of our information browser. We mentioned that we suspected a higher cognitive load to be a possible reason for this resistance to switching. This load may, in part, be a result of our user's inexperience with wayfinding in an information space through auditory feedback. Certainly, the successful mode switching reported by Suhm et al. (2001) with the graphical user interface suggests this possibility. Finally, our sighted users were performing a task that required them to listen to and comprehend the information being read by the text to speech engine. It may be that the output modality influenced the choice of input modality, that is, users stayed with the speech mode because the dialogue between the human and the computer system suggested a real conversation. Thus, the natural reaction was to continue speaking.

FURTHER RESEARCH

The present study has reported the results obtained from sighted participants when using integrated speech and tactile inputs. These results provide a baseline for understanding the difference between sighted users and blind and visually impaired users in using multimodal interaction. This is a preliminary step toward learning how to combine different input modes into a strategically integrated whole non-visual system. Further research will be conducted with blind and visually impaired participants following the same experiment procedures on the same system to obtain comparative results.

REFERENCES

1. Halverson, D., Horn, D., Karat, C., and Karat, J. The Beauty of Errors: Patterns of Error Correction in Desktop Speech Systems, in *Proceedings of INTERACT'99*, Edinburgh, September 1999, IOS Press, pp. 133-140.
2. Karat, C., Halverson, C., Horn, D., and Karat, J. Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition systems, in *Proceedings of CHI'99*, Pittsburgh PA, May 1999, ACM Press, pp. 568-575.
3. Oviatt, S.L. Multimodal Interfaces for Dynamic Interactive Maps, in *Proceedings of CHI'96*, Vancouver, Canada. ACM Press, New York, 1996, pp 95-102.
4. Oviatt, S.L., Ten Myths of Multimodal Interaction. Communications of the ACM, November 1999, Vol. 42, No. 11. 74-81.
5. Oviatt, S.L. Mutual Disambiguation of Recognition Errors in a Multimodal Architecture. In Proceedings *CHI'99*, May 1999, Pittsburgh, PA. ACM Press, NY, pp. 576-583.
6. Oviatt, S.L. and Cohen, P. Multimodal Interfaces that Process What Comes Naturally. *Communications of the ACM*, March 2000, Vol. 43, No. 3, 45-53.
7. Oviatt, S.L., Laniran, Y., Bernard, J. and Levow, G.A. Linguistic Adaptations During Spoken and Multimodal Error Resolution, *Language and Speech*, 1998, 41(3), 419-442.
8. Oviatt, S.L. and Olsen, E. Integration Themes in Multimodal Human-Computer Interaction, in *Proceedings of the International Conference on Spoken Language Processing*, 1994, pp. 551-554.
9. Oviatt, S. and VanGent, R. Error Resolution during Multimodal Human-Computer Interaction, in *Proceedings of the International Conference on Spoken Language Processing*, 1996, pp. 204-207.
10. Sears, A., Feng, J., and Oseitutu, K., Hands-Free, Speech-Based Navigation during Dictation: Difficulties, Consequences, and Solutions, *Human-Computer Interaction*, 2003, Vol. 18, pp. 229-257.
11. Suhm, B., Myers, B, Waibel, A. Multimodal error correction for speech user interfaces, ACM Transactions on Computer-Human Interaction, Vol. 8, No.1, March 2001, pp. 60-98.
12. Williams, C., Tremaine, M. SoundNews: An Audio Browsing Tool for the Blind. In *Proceedings of the International Conference on Universal Access in Human-Computer Interaction (UAHCI),* August 2001, pp. 1029-1033.