SpeechToast: Augmenting Notifications with Speech Input Focus

A.J. Bernheim Brush and Paul Johns
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
{ajbrush, pauljoh}@microsoft.com

ABSTRACT
To explore the value of speech input focus for handling notifications, we built and deployed SpeechToast, an Outlook Add-in that replaces standard email notifications with a version that includes speech input commands (e.g. "open", "delete"). Notifications shown by SpeechToast have speech input focus when the audio context surrounding the computer is favorable for speech recognition. We deployed SpeechToast to 18 current users of email notifications for 4 weeks. Overall, speech input focus appealed to some participants, while non-users indicated their willingness to have it enabled as long as it did not detract from their experience. Our research suggests that selectively enabling speech input focus could provide natural and intuitive interactions that complement other input modalities.

Categories and Subject Descriptors
H5.2: User Interfaces.

General Terms
Design, Human Factors.

Keywords
Notification, speech, speech input focus, field study.

1. INTRODUCTION
Email notifications, instant messages, calendar alerts, telephone calls and other interruptions are part of knowledge workers daily lives [e.g. 7]. Researchers have explored a variety of ways to reduce the potential cost of interruptions including using task structure to predict interruption cost [5], predicting human interruptibility using sensors [2], prioritizing delivery of notifications based on their inferred importance [4] and changing presentation of the notification based on utility [6].

In contrast to approaches that seek to change when, what, and how notifications are delivered, we wanted to explore notifications handling. We hypothesized adding a speech input focus, distinct from the standard window focus, that allowed users to handle notifications using speech without needing to move hands involved with typing on a keyboard or using the mouse could make handling notifications feel more efficient and less distracting. To explore this hypothesis, we built SpeechToast, an Outlook Add-in that replaces the existing Desktop Alert notifications displayed by Outlook when new mail arrives (see Figure 1). SpeechToast notifications are automatically enabled with speech input focus when the audio context surrounding the computer is favorable for speech recognition. The user can then speak commands to interact with the notification (e.g. open, delete, reply) rather than click.

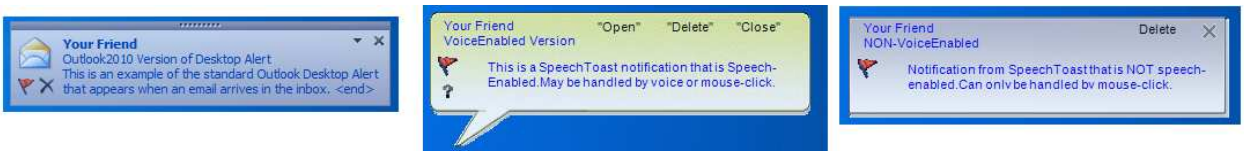
We deployed SpeechToast to 18 users of Outlook Desktop Alerts in a 4 week field study that alternated between a non-speech and speech-enabled condition. Participant preference for using speech varied based on the degree they typically interacted with notifications, their personal speech recognition experience, and comfort speaking to their computer. Some participants were very enthusiastic, while others found using speech uncomfortable.

Analogous to how keyboard shortcuts provide value to some users without detracting from the experience of others, non-users indicated their willingness to have speech input enabled if it did not detract from their experience. While the recognition issues some of our participants experienced must be addressed, most participants had relatively few speech recognition problems showing that speech input focus is technically feasible. We believe SpeechToast demonstrates the potential of selectively enabling speech input focus to provide natural and intuitive interactions that complement other input modalities.

2. SPEECHTOAST OUTLOOK ADD-IN
To gather initial data about whether users would find speech input focus appealing for handling notifications, we conducted a lab study with 12 Outlook users from outside our company. Participants filled out Mad Libs (filling in the blanks in a story using funny words) while handling email notifications and meeting reminders with speech, mouse, and then in a "free choice" section where they could use either speech or mouse. Words they needed to collect to finish a task occasionally appeared in the notifications, to keep them paying attention to the notifications.

Lab study results were encouraging. Eight of twelve told us they preferred using speech to handle email notifications, and during the free choice section they used speech to handle the email notifications 88% of the time. Participants reported feeling they were saving time using speech (e.g. "I didn't have to stop what I was doing and drag the mouse over that way.") and their work flow was less interrupted (e.g. "I had never used voice recognition [previously], but found it to be useful while working on a task and not having to stop my task to take action.").

AVI '12, May 21-25, 2012, Capri Island, Italy Copyright © 2012 ACM 978-1-4503-1287-5/12/05…$10.00.



(a) **(b)** **(c)**

Figure 1: Example of Outlook2010 Desktop Alert Notification (a) and SpeechToast Notifications when speech enabled (b) and not speech enabled (c).

Inspired by the study results, we built the SpeechToast Outlook Add-in to assess the value of speech input focus for notification handling outside the lab. SpeechToast replaces the current Desktop Alert feature in Outlook which shows a small notification window in the bottom of the main screen when new email arrives in the Outlook Inbox (see Figure 1a). The SpeechToast speech enabled notifications (see Figure 1b) provide the same functionality as the default notifications: "Open", "Close", "Delete", "Flag" and "Mark Read", either by speaking or clicking on the commands. We also enabled additional speech-only commands: "Reply", "Reply All", "Forward", and "Help". Consistent with the Desktop Alert behavior, the SpeechToast notification displays on top of other windows, but does not take keyboard focus from the current application.

We carefully considered three of the challenges outlined by Bellotti et al. [3] for sensing systems in our design: *Attention* (how I know the system is ready and attending to my actions), *Alignment* (How do I know the system is doing the right thing) and *Accident* (how do I avoid or recover from errors). To indicate when a notification had speech input focus ("Attention"), we used the "speech bubble" window shape, yellow background, and put command words in quotes. Mischke has also described the importance of making clear to users when speech input is available [8].

If the user speaks a command, SpeechToast gives feedback to show what happened ("Alignment"). For most commands, the resulting action gives the feedback (e.g. a message opens after a successful "open"). For other commands, SpeechToast turns the command red and underlines it to give visual feedback. In addition, to minimize the ramifications of a false positive "Delete" speech recognition event we created a 'Voice Deleted' folder in Outlook where messages that SpeechToast deletes are placed for easy review.

In our initial prototype, SpeechToast immediately showed the speech enabled notification when a new message arrived in the inbox. In our own use, we had an unacceptable number of false positive speech recognition events. To avoid these "Accidents," we modified SpeechToast to "listen" for 4 seconds when a new message arrives before showing a notification. If SpeechToast recognizes any speech during this period, the add-in decides that the user's environment is not currently conducive to using speech (e.g. two people having a meeting) and displays a blue-gray non-speech enabled clickable notification similar to the standard Outlook Desktop alerts (see Figure 1c). In either case, the notification disappears after 10 seconds with no interaction.

SpeechToast utilizes the Microsoft Windows 7 Speech Recognition Engine. The speech grammar contains the 9 commands mentioned above and a garbage collection word to enhance command recognition. Based on initial experiments we set a minimum speech recognition confidence level of 0.94. For research purposes, SpeechToast logged all notification events and saved the 2 second audio stream that triggered each speech recognition event.

3. FIELD STUDY

To evaluate SpeechToast, we recruited participants using Desktop Alert notifications so we could study participants' experience with speech input focus without introducing new unfamiliar interruptions. We deployed SpeechToast to 18 people (9M, 9F) in a four work-week field study with two conditions: Speech Mode (SM) and Non-Speech mode (NS). Participants spent the first week in NS and used a non-speech enabled version of the add-in (all notifications resembled Figure 1c). Next, participants used SpeechToast with speech enabled when the audio environment supported it for two weeks in the SM condition. Participants returned to the NS condition for the last week of the study so we could ask them whether they missed speech to address any potential novelty effects. All participants had their own office, and are fluent English speakers. Participants received $25 at our company café and were entered in a lottery for a $100 gift card.

We collected metrics about the messages received, whether participants ignored or handled notifications, and interviewed participants at the end of each condition. Total days in study for each participant was around 20 (Avg: 21) and varied slightly due to scheduling issues and unanticipated vacations. To compensate for different study lengths we primarily report percentages. We also found that many participants left Outlook running overnight which meant notifications might be shown and logged that they did not see. For consistency we report metrics on notifications received from 8 am to 8 pm on weekdays.

For the best speech recognition experience possible we provided Microsoft LifeCam Studio 1080pHD web cameras to 14 participants that did not already have web cameras on their computer or ones with lower quality microphones. We verified the quality of the microphone for the other 4 participants. Participants completed the 15 minute standard Windows Speech Recognition training to improve recognition accuracy. To train participants on the SpeechToast commands, we sent them an email with the command in the subject line twice for each command (18 total). Although we piloted SpeechToast extensively ourselves and with colleagues, participants occasionally reported the notification window stealing keyboard focus. We fixed bugs as they were found and then provided updates to the affected participants.

4. RESULTS

Our participants received 16,326 messages during the study period. Average messages received per day varied from 14 to 90 (median 40). Logging during the first non-speech week demonstrated that the typical behavior of our participants was to interact with only a small fraction of their notifications (6%). We classified our 18 participants into 3 groups based on how they interacted with notifications during the study using log data and participant comments. Eleven participants were *Speech Users* who

Table 1. Notification handling by different types of users. The Non-Speech columns include all notifications shown in Non-Speech mode and non-speech notifications shown in Speech Mode weeks when the environment was unfavorable for speech.

	Non-Speech Enabled Notifications (Figure 1c)		Speech Enabled Notifications (Figure 1b)			
User Group	**Received**	**Handled**	**Received**	**Handle with Mouse**	**Handle with Speech**	**Total Handled**
Speech Users (11)	6129	331 (5%)	3357	220 (7%)	142 (4%)	362 (11%)
Mouse Users (3)	1463	105 (7%)	1409	163 (12%)	0 (0%)	163 (12%)
Read-only (4)	2155	2 (0.1%)	1813	4 (0.2%)	4 (0.2%)	8 (0.4%)
Total (18)	9747	438 (5%)	6579	387 (6%)	146 (2%)	533 (8%)

interacted with notifications using speech and the mouse. Of the remaining seven participants, three participants were *Mouse Users* who interacted with notifications only using the mouse. The final four participants were *Read-only* users who did not typically click on any notifications. Table 1 shows the differences in notification handling by the three user groups.

4.1 Speech User Participants Handled More

The eleven Speech User participants handled a higher total percentage of notifications when speech was enabled (Table 1, first row, 5% vs. 11%). Nine of the 11 Speech Users increased the number of notifications handled ranging from an 11% to 301% increase, with a median 116% increase. Handling emails when received (e.g. by deleting or responding) reduces the number to process later, a "one touch" strategy recommended by some productivity consultants (e.g. [1]). For the 362 speech-enabled notifications Speech Users handled, participants used speech 39% of the time. This percentage was 40% in week 1 and 38% in week 2, showing basically no novelty effect in use of speech.

Overall, "Open" was the most common command spoken (49% of all speech input commands), followed by Delete (21%), and Close (18%). These speech commands were consistent with the Non-Speech weeks, when participants clicked mostly on "Open" (82%), and less often on Delete (10%) and Close (8%). However, the availability of speech seemed to encourage using of commands besides Open and some participants took advantage of the Reply command (10%), which is not available in traditional notifications.

4.2 Efficient for Some

Our hypothesis was that participants might perceive interacting with notifications using speech to be more efficient. The eleven Speech Users were most positive about the efficiency of speech input and the median response was "Agree" that handling notifications using speech felt more efficient than using the mouse (5 pt. Likert scale from Strongly Disagree to Strongly Agree). For example, P1 commented "handled more, easier to say open than to stop typing and say open, easier to not take my hands off [the keyboard]" and P9 said "Truly it is more efficient when you don't have to switch back and forth [between keyboard and mouse]."

Comparing whether participants chose to use speech or mouse when they interacted with notifications (when both were available), we found that six participants used speech more than half the time and the median overall was to use speech for 57% of interactions. Some of these participants were quite enthusiastic about the addition of speech input, commenting 'It went great. I love it. Totally works." (P4), and "Like the voice" (P1).

4.3 Mental Effort Can Be Distracting

On the other hand, Speech User participants were "Neutral" about whether using speech input was less distracting, although 4 participants agreed. Some participants described additional mental effort necessary to use speech input. P2 said "it actually is more distracting, because with voice you need to focus on trying to answer it before it disappears." P3 commented "there were times where I would look it, I wouldn't mind opening it. But turning it into words was mental effort." Another issue was unfamiliarity of speech, mentioned by two of the three Mouse User participants. P5 felt he used the mouse because it was familiar and speech input was not "primed." Similarly, P11 told us "when I'm working my brain is just wired to clicking." Speech recognition errors also appeared to contribute to an increased sense of distraction. P14 commented "When I say "open" and it doesn't work it's really frustrating and then I have to go and click."

Six participants also commented during interviews that speaking to their computer felt awkward. For example, P7 said "It was weird for me to be talking when it was quiet." This awkwardness was enhanced when nearby colleagues made comments. For example, P17 told us SpeechToast was fun to use, but people could hear her down the hallway. P6 said her neighbors asked "what are you doing, that is kind of weird" and P1 was relieved to discover he could speak pretty quietly. P12, who only used the mouse, commented that speech felt unnatural since he typically sits coding quietly all day.

4.4 A Reasonable Alternative

Immediately after the Speech Mode condition, we asked participants what modality they preferred for notifications they were going to handle. Of the Speech Users, 2 preferred speech, 3 had no preference between mouse and speech and 4 told us they preferred mouse (2 answered other). While there was less preference for speech input than we hoped, several participants told us that they would be fine having speech input focus enabled so that others could use it, as long as it did not distract from their notification handling experience (e.g. no false positives). P3, who preferred mouse, commented "having it [speech input] as an alternative seems totally reasonable."

Absence also appeared to make participants grow fonder of SpeechToast. After the final Non-Speech week, the median across all participants was "Agree" that they missed speech input for handling notifications. When asked if they wanted to return to using SpeechToast, 6 participants said "yes" and 6 said "maybe." Many of the people who answered "maybe" told us they were indicating their desire for speech input, but required an improved version of the prototype for continued daily use. Most wanted

improved speech recognition, e.g. P9 told us "Need better speech to work."

Some participants also reported that they wanted the email message window enabled with speech input focus to better support their entire notification handling process. For example, P1, who was an enthusiastic user of speech input, told us after opening a message using speech he would like to speak a command (e.g. "Close", "Reply", "Delete") once he finished reading the message. Similarly, P15 who almost never used speech input, explained if he "can't close [a message window] with speaking - I might as well leave my task mentally" and use the mouse to open the message from the notification window.

4.5 Environment was Suitable

During the field study we also wanted to evaluate the suitability of participants' offices for speech input. We recruited participants with their own offices since speech input is clearly not appropriate for all settings (e.g. open plan offices). We found for the majority of our participants the noise level in their office was conducive for using speech input. Overall, the median percentage of speech-enabled notifications during the Speech Mode condition was 94% and eleven participants had more than 92% speech enabled notifications. P9, the participant with the noisiest environment still had 56% of her notifications speech enabled.

Examining how well speech recognition worked for participants, their median response was that SpeechToast "rarely" (< 24% of time, 6 point scale) completely failed to recognize their voice command. However, three participants reported more than 75% of the time speech recognition completely failed for them. Investigating this further, we believe most problems related to work styles that resulted in frequently plugging and unplugging of their microphone, which meant SpeechToast did not always have access to it.

Participants also reported false positives (when Speech Command executed a command they did not say) happened "rarely" (< 24%, median response) and 8 people reported no false positives. However, three participants reported several false positives, including the Help Dialog opening due to keyboard noise. We reviewed the logs of these participants and modified 101 incorrect entries (1.5% of all speech-enabled notifications), changing them from "handled-by-speech" to "ignored." We also found more instances of the newly available speech commands (e.g. "Reply All") than expected. We reviewed audio logs and removed false positive entries. All data previously reported uses corrected numbers. Lastly, when asked to review their VoiceDeleted folder, 16 participants found no problems and two (who had microphone issues) reported a few, but did not perceive it as a big problem.

Thus, while recognition worked reasonably well for most, there were some participants that had challenges and this was reflected in their qualitative responses. Overall, we believe speech recognition worked well enough that participants could experience speech input focus and give useful feedback about its appeal to them. Moving forward, while we do not underestimate the challenge of robust speech recognition using desktop microphones, we anticipate that many of the problems experienced by our participants could be mitigated by improved built-in microphones and additional refinement of the speech recognition engine for our use case (we used the default windows recognition engine primarily designed to support dictation).

5. CONCLUDING REMARK

Building and deploying the SpeechToast Outlook Add-in allowed us to experiment with enabling a separate speech input focus in addition to the standard keyboard and mouse focus. The appeal of using speech to handle notifications varied for our participants depending on their approach to notification handling, their personal speech recognition rate (good for many, bad for some), and comfort speaking to their computer. Reactions ran the gamut from very enthusiastic use of speech input to non-use.

Much as keyboard shortcuts provide value to some users without detracting from the experience of others, we believe speech input focus provides a natural and intuitive interaction that complements other input modalities. While we explored speech input focus for Outlook Email notifications, we believe this approach can be generally applied to many other interruptions, such as calendar reminders, IM announcements, and other interruptions with limited response choices. While remaining technical problems must be addressed to provide consistent speech recognition experience for all participants, our research demonstrates the appeal of speech input focus to a segment of the population, particularly for notification handling.

6. ACKNOWLEDGMENTS

We thank the people that participated in the study.

7. REFERENCES

[1] Allen, D., 2002. *Getting Things Done: The Art of Stress-Free Productivity*. Penguin, NY, NY.

[2] Avrahami, D., Fogarty, J., Hudson, S., Biases in Human Estimation of Interruptibility: Effects and Implications for Practice. CHI 2007, ACM Press (2007), 51 – 60.

[3] Bellotti, V., Back, M., Edwards, K.E, Grinter, R., Henderson, A., and Lopes, C., Making Sense of Sensing Systems: Five Questions for Designers and Researchers, CHI 2002, ACM Press (2002), 415-422.

[4] Horvitz, E., Jacobs, A., Hovel, D, Attention-Sensitive Alerting, UAI '99 ACM Press (1999), 305 – 313.

[5] Iqbal, S., & Bailey, B., Leveraging Characteristics of Task Structure to Predict the Cost of Interruption, CHI 2006, ACM Press (2006), 741- 750.

[6] Gluck, J., Blunt, A., McGrenere, J., Matching Attentional Draw with Utility in Interruptions, CHI 2007, ACM Press (2007), 41 – 50.

[7] Gonzalez, V., and Mark, G., "Constant, Constant, Multi-tasking Craziness:" Managing Multiple Working Spheres, CHI 2004, ACM Press (2004), 113 – 120.

[8] Mischke, A., Multimodal UI Guidelines for Mobile Devices, In Meisel, H. (ed.) Speech in the User Interface: Lessons from Experience, pp. 219 – 224, Trafford Publishing, 2010.