RC23182 (W0404-082) April 14, 2004
Computer Science

IBM Research Report

e-PIM: A Conversational Multimodal Interface for a Thin Client

Jennifer Lai, Stella Mitchell, David Wood, Christopher Pavlovski, Harry Stavropoulos
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

e-PIM: a Conversational Multimodal Interface for a Thin Client

Jennifer Lai, Stella Mitchell, David Wood, Christopher Pavlovski, Harry Stavropoulos
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
jlai, cleo, dawood, hstavrop @us.ibm.com, chris_pav@au1.ibm.com

ABSTRACT
As Third Generation (3G) networks emerge they provide not only higher data transmission rates but also the ability to transmit both voice and low latency data within the same session. This paper describes the successful implementation of a multimodal application (voice and text) that uses natural language understanding combined with a WAP browser to access email messages on a telephone handset. We also report on a user trial that evaluated both the multimodal system and a unimodal system that is representative of current products in the market. Participants saw significantly greater value in the multimodal interaction, and rated their experience with the multimodal system significantly more positively than the unimodal system. They were also significantly faster and more inclined to use and recommend the multimodal system. While we expected to see mixed usage of modalities in the multimodal system, speech was the dominant modality used, with users falling back to GUI selection only after encountering multiple speech recognition failures in a row. To our knowledge, this represents the first implementation and evaluation of its kind using this combination of technologies.

KEYWORDS: Multimodal interfaces, natural language understanding, mobile phone, speech technologies

INTRODUCTION
Multimodal interfaces, i.e. interfaces that accept at least two input modes, have really only started to be used and seriously researched in the past 20 years [10]. Multimodal systems present a significant advantage over unimodal systems in that they are accessible by users with a wide range of capabilities and are usable in a variety of environmental settings. They are often viewed as the solution to increasing the robustness and accuracy of speech-only systems [10] and appear to be well suited for use in a mobile computing environment given the varying constraints placed on both the user and the recognition technology by the mobile device's small keyboard and screen, the wireless network audio encoding quality and the background noise present in the user's environment. The additional modality can be used either in a complementary manner (to supplement the recognition technology), in a redundant manner or as an alternate to the recognition technology in case of high error rates or a context of use that does not support one of the input modes. Multimodal systems often combine more than one form of output as well, creating multimedia effects by using visual and auditory output.

Prior research has shown that users' multimodal utterances are briefer, contain fewer complex descriptions and have as few as half the number of disfluencies as speech-only input [8]. This same study also found that users had a strong tendency to switch modes after a recognition error thus leading to smoother error recovery.

Creating a highly usable interface for browsing and accessing large amounts of textual information over a mobile phone represents a substantial challenge. It is not unusual for the average knowledge worker to receive over a hundred messages a day. The sequential navigation model that was established for voicemail messages, and that is still applied to many unimodal mobile email systems today, does not work well when applied to large numbers of messages. When viewing the daily onslaught of messages with a graphical user interface (GUI) users routinely do a visual triage, scanning for messages from people whom are important to them, or for message topics that pique their interest. This triage is difficult to do when relying on auditory output, which is slower than the visual channel. A multimodal interaction model thus appears to be ideally suited to mobile email retrieval because it supports visual browsing, and the combination of modalities can help to improve the robustness of the speech recognition in the very challenging environment of mobile usage.

This paper describes a multimodal implementation for a mobile email retrieval application in a 3G network environment, along with the results of a user trial that was conducted to evaluate the system. The trial evaluated the usability of the multimodal system and compared it to an existing unimodal system in the same domain. The user study was conducted in Sydney in order to take advantage

of the 3G networks that support the transmission of voice and data within the same session. The implementation used an off-the-shelf, unmodified device. To our knowledge, the e-PIM system is the first fully functioning implementation of a conversational multimodal interface on an unmodified telephony device.

PRIOR WORK

For most domains, the preferred modes of interaction have yet to be established, or may be heavily dependent on personal preferences and history of success, as well as context of use. However, in spatial domains such as interacting with a map, research has shown that users prefer speech for describing objects and events, as well as for issuing commands. Preference for pen input increased when input required digits, words in foreign languages, symbols or locations [6]. Efficiency gains have also been measured in this domain, with 10% faster completion times for visual-spatial tasks when using multimodal pen/voice interaction, although there was no impact for verbal or quantitative tasks in this particular study [6].

The most mature combinations of input modalities involving a recognition technology are speech and pen input (e.g., [9]), as well as speech and lip movement [5]. Systems that include some form of vision processing of gestures or facial expressions are maturing and attracting more attention as they mature. Vision-based systems allow for monitoring of the user and thus are useful for pre-qualifying the context of subsequent interactions [4, 14]. For example if the user directs his gaze towards the air-conditioning and requests "turn it up please", the system can use contextual information to create a more successful outcome.

Additionally, given the impoverished interface of a cell phone, researchers have examined ways of enriching the user experience by adding tilt sensors, (e.g. [12, 11, 16]) touch and proximity sensors (e.g., [1]) as well as eye contact sensors [15].

E-PIM

The e-PIM (electronic- Personal Information Management) project focuses on facilitating mobile communication by providing multimodal access (natural language speech and graphical browser) to enterprise email and calendar entries from a cell phone. User input can be in the form of text input, GUI selection, or speech recognition. Output is a combination of written text and spoken synthetic speech.

Functionality

Supported functions include reading, sending, forwarding, replying, summarization and deletion of email messages, as well as checking calendar entries, and creating appointments. By editing a personal profile, users can set their password and configure certain aspects of the application, such as how many days worth of email messages to retrieve, how to pronounce their name, or how fast the text-to-speech voice should speak.

Users can interact with e-PIM using both voice and the GUI browser: requests are made from either modality and the system response is presented on both modalities at the same time. Although all the core capabilities are available in both the audio and visual interface, each modality has certain traits that are only available in that modality. For example, using a spoken command the user can request messages about a specific subject (e.g., "do I have any messages about the seminar"), whereas using the GUI browser the user can only request a listing of all the messages in their inbox. During email creation, the graphical browser provides richer function allowing any recipient name to be entered by text, while the spoken interface only recognizes the set of names consisting of other e-PIM users, personal address-book contacts and senders of messages listed in the inbox. While these differences are valuable to the user because they capitalize on the strength of each modality, communicating the functionality to the user is a design challenge.

Voice Interface

Most speech-based telephone interfaces on the market today use a directed style of system prompts in which users are presented with menu options from which they can make a selection; navigating in a controlled manner until the task is completed. Much of the naturalness and power of speech is undermined when the application relies too heavily on the use of directed dialogs, and the user can feel confined to the passive role of waiting for the system to prompt for a specific answer. A Natural Language interface allows the user a higher degree of freedom with less cognitive load since there are no commands to memorize or hierarchies to navigate. E-PIM's voice interface employs a natural language understanding technology [2] that uses statistical techniques to transform text generated by the speech recognizer into formal language statements that express the meaning of the utterance. This allows the user to be very open with their vocabulary and phrasing. The system also supports mixed-initiative dialog, which means that users can switch to a new task without completing a task that they previously initiated. For example:

User: set up a one hour meeting tomorrow
System: what time should the meeting start?
User: do I have any messages from David Smith?
System: you have three messages from David Smith

Messages in the Inbox can be accessed using a number of parameters including date, sender, subject keyword, urgency, and ordinal number (e.g., "read me the second message from David Smith", or "show me all the urgent messages received yesterday"). In order to facilitate the presentation of long messages either by speech, or on a small screen, users can request a summary of the message.

Calendar entries can also be queried via a number of parameters including, time, time range, date, date range, type, and host or invitee (e.g., "Show me my calendar entries from 2 to 4 pm tomorrow")

In addition to the core functions described above, the interface supports time and date queries along with some additional requests that help maintain a productive dialog such as:

• **Guide me**: drops back to a more directed style of dialog to help walk the user through his choices;
• **Help**: presents contextual help
• **Repeat**: repeats the last system response
• **Cancel**: aborts the current operation

Visual Interface

e-PIM's visual interface provides four primary choices (see Figure 1) on the main menu screen. These correspond to the core functions that are also available through the voice interface. The one exception is the calendar entry creation function which was only fully available through the voice interface at the time of the pilot and was not part of the user trial.

Figure 1. e-PIM main menu

The user activates a link by navigating to it and selecting it through use of the four-way scroll button on the phone. For most requests, the response is displayed all at once, with scrolling used as necessary to access all the information. However, in order to best display the "show email" response on the small screen, the email headers only contain sender and subject (see Figure 2). The body of the message is viewable by selection, with additional details (e.g. date and time of the message) being available by click-through.

For interactions that support text entry such as the "send email" screen, if the user decides to enter text rather than using voice, the text is entered by using the keys on the telephone keypad with the multitap method. This method requires a user to hit the "2" key twice for the "b" character for example. In addition several common subject lines and brief email replies ("yes, sounds fine") are provided as drop down list selections from the GUI.

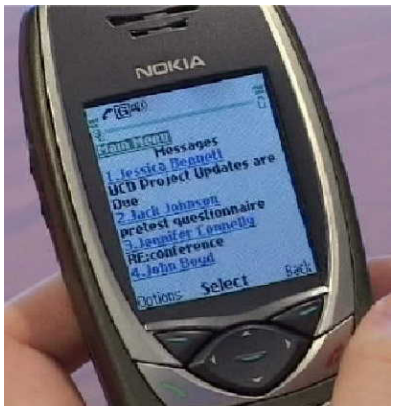

Figure 2. Screen shot of the "show email" GUI display

Multimodal Synchronization

Synchronization behavior must be defined both for input (the way in which input from separate modes is combined) and for output. The W3C [17] distinguishes several types of multimodal synchronization for input as follows:

- **sequential**: two or more input modalities are available, but only a single modality is available at any given time.
- **simultaneous**: allows input from more than one modality at the same time, but each input is acted upon separately in isolation from the others.
- **composite**: provides for the integration of input from different modes into one single request.

The e-PIM system uses simultaneous synchronization. Each spoken or GUI submission is treated as a complete input operation by the application. A submission from the voice mode is one utterance - sometimes a free form request such as "do I have any urgent messages from Tom" and sometimes a short answer (e.g. "no"). A submission from the GUI occurs when the user clicks on a link that sends a request to the application. Input from the different modes is not currently combined into a composite request. This means for example, that the user cannot say "read me this one" with the second email on the display selected and have the system resolve the dyadic reference. In addition, in this first version of a multimodal e-PIM, we did not address the issue of simultaneous conflicting input. This did not prove to be a problem during the trial, however for longer term rollout this type of error prevention would have to be addressed.

e-PIM's output synchronization is best described as form-level, which is defined by the W3C as "all modalities are updated only at certain application defined points in the interaction" [17]. Each active modality is updated on each submission from the user. However, delivering a synchronized result to the user proved difficult in a real deployment due to differences in latency times between the

circuit-switched service and the packet-switched service. (For details on packet and circuit switched services see [13].) The system response was typically presented to the user on the voice channel slightly before it was presented visually. To reduce the impact of the problem we made sure to always send the visual content prior to sending the voice content. Users in the study did not mention this slight delay between the voice and the text when discussing their impressions of the system.

Device

We decided that the solution developed had to be independent of the client device in order to eliminate the need to manage software distribution and to facilitate larger scale deployments by supporting a variety of standard devices. We piloted e-PIM on a Nokia 6650 mobile phone, which is a class A device (i.e. a device that supports simultaneous circuit-switched and packet-switched connections.) The device screen is full color with 128 by 160 pixels, four way scroll and user changeable font size. The Wireless Markup Language (WML) browser on the device is a single threaded phone-based browser supporting WML 1.3. The device supports Service Indication (SI) Wireless Application Protocol (WAP) push but not Service Loading (SL) WAP push. A SI contains a short text message and a URL; if the user accepts the SI, content is fetched from the URL. In SL a user agent on the device can process the push message and fetch content from the URL without user intervention. Browser support for Service Loading was not available at the time of the pilot on the device we used. Since we did not want the interaction to require user intervention for each GUI update resulting from voice input, we used WAP push only in establishing the initial connection between the GUI browser and the application. For subsequent pushes we adopted the polling approach described in the architecture section below. Since the browser is single-threaded, the user is prevented from inputting to the GUI when it is occupied by using a blocking request to a polling service.

Architecture

Third generation networks provide 'multicall' capability that enables concurrent connections from the mobile phone to both voice (circuit-switched) and low latency data (packet-session) networks. We leverage the multicall supplementary service capability of 3G networks in order to support simultaneous voice and GUI interaction. The circuit-switched network is used to establish the voice call between the mobile device and the telephony platform. An SMS channel in the circuit-switched network is used to establish the initial connection between the application server and the visual browser on the phone. The packet-switched network is used to transport WML content over a WAP stack between the mobile device and the WAP gateway.

In order to support the voice interface, speech recognition and synthesis capabilities must be present either on the device or in the network. Speech embedded in the device has limited capability compared with speech resident in the network. Network-based speech for example supports statistical language model-based recognition which can theoretically recognize any sentence constructed of words in the model's vocabulary. This type of recognition is generally used to support natural language interfaces [2]. Speech embedded in the device may allow recognition of only a few hundred words uttered in a structured format. Embedding speech technology into the device also places further dependencies on the product development lifecycle of handsets, with little guarantee of consistency of user experience across a range of handsets. As a result of these considerations, all of the speech processing for the e-PIM system is done on the telephony platform server located within the network. Figure 3 shows the logical components of our solution architecture.

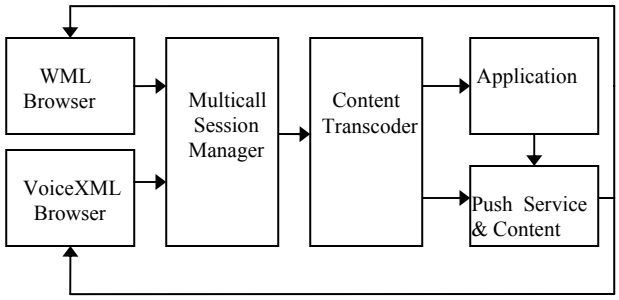

Figure 3. multimodal e-PIM logical architecture

The VoiceXML 2.0 client browser is located on a server within the network, and it is extended with two capabilities. The first is the ability to have content pushed to it asynchronously (to support multimodal interaction) and the second is support for statistical language model based recognition (to support natural language understanding in the speech interface). Each VoiceXML document dynamically generated by the application for the client browser contains one prompt, and collects one input field, which is the user's utterance.

The WAP client browser used in this e-PIM deployment was a standard phone-based browser. The initial push to the WAP browser is accomplished by sending a Service Indication message to the calling device. After that, a polling approach is used to deliver pushed information to the visual client. A URL referencing the pushed document for the next dialog turn is included in all content prepared for the WAP browser. The URL is incorporated into a WMLScript embedded in the final markup delivered to the browser. After the page is loaded, the WMLScript issues a blocking request for the document to the push service content server, which polls for the file and only returns if it is found.

The multicall connection manager uses cookie management to maintain the association between different connections (voice and data) from the same device, and it modifies incoming requests so as to present a single client to the application manager. At runtime the content transcoder transforms the application response into a markup suitable for the client browser on which the response will be presented. We implement this by using an XSLT engine with a style sheet for each supported client. Currently there are three supported clients: an enhanced VoiceXML 2.0 browser for the voice interface; a WML 1.3 browser for the mobile phone GUI implementation; and HTML 3.2 for the GUI implementation on a phone-enabled iPAQ.

The application provides natural language support for the speech interface, multimodal dialog management, and the core PIM functions. In response to each request the application returns a compound document, which contains content for both the voice and the visual interface. There is one application instance associated with each active user.

Interaction Flows

Starting the application

The user starts the application by calling the phone number of the system. As e-PIM answers and begins speaking the welcome prompt, the user is notified that he has received a Service Indication. If he accepts the Service Indication by clicking on it (a "click" is accomplished by pressing the selection key on the phone with the item highlighted), the login page is displayed on the phone screen (see Figure 4). At this point, the user can either say his/her name, or type it into the GUI browser using the telephone keypad.

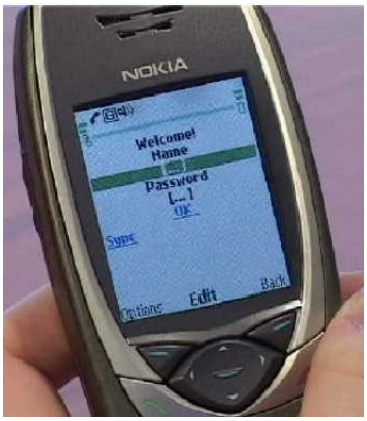

Figure 4. e-PIM login screen

Multimodal interaction

We did not want the voice to simply mirror the text from the visual display, and vice a versa. Since it is an NL system, the voice interface uses open ended prompts (e.g., "what next?") or asks for a specific piece of information related to the current task (e.g., "what is the start time for this meeting?"). On the other hand, the GUI presents either the main menu of choices or the complete form (with all the input fields) associated with the current task. When presenting the results of a user's query, the output modalities usually present the same content, but sometimes just a summary is given by voice ("you have 21 new messages") with the details being presented visually (a list of headers appears on the screen).

When a user's utterance is not recognized or when the user has been silent for too long, the voice interface provides feedback about these events. In these cases, even though it is a multimodal application, we do not update the GUI and instead simply let the user try again with the original information available. In this manner, the modalities are used in a complementary fashion, rather than a redundant one. In the current implementation, the voice and text are redundant when the user either asks to have a message read to him, or selects the message body from the GUI. In either of these situations the same information is presented both by voice and text. As mentioned earlier, the voice starts slightly before the text is displayed. Thus if the user is interacting with the device using the GUI and wants to just read the text, he still has the text read to him. As it turns out this was one of the aspects that users complained about in the study.

Primarily because the browser is single threaded, GUI input is only possible for a short window of time following a GUI refresh. When a new page is loaded, there is an application-configurable period of time during which the user can either select a link or begin text entry (they do not need to complete text entry). A spinning globe in the upper right corner of the display indicates that polling has been activated and GUI input is disabled. Once the globe starts to spin, users are unable to interact with the GUI until the display is refreshed (as a result of spoken input) or they interrupt the polling by selecting cancel. We found that a window of 7 to 10 seconds worked best since polling is typically not needed before that, and it allows a user enough time to navigate the GUI and initiate action.

Ending the application

It is left to the user to terminate each mode independently. They can hang up the voice session and keep interacting through the GUI, or end the WAP session and proceed with a voice-only call. Only after both modalities are terminated, or a timeout period of inactivity passes, is the user session ended.

STUDY

A user study was conducted with a fully functioning multimodal email system in a 3G environment. The goals of the study were to:

1. Test the prototype implementation with representative users;
2. Measure the incremental value of the multimodal system compared to a unimodal solution;
3. Determine users' response to the multimodal system along with their willingness to buy or recommend;

4. Obtain objective performance metrics for identical tasks with both a unimodal and multimodal system

In order to measure the value of the additional modality (speech), we created a baseline measurement for each participant by having him/her use a unimodal system for accessing email. The unimodal system was selected as being representative of systems that use a WAP browser for mobile mail access (see Figure 5). The same physical handset (the Nokia 6650) was used to access both systems. (The unimodal application can not be identified at this time for reasons of client confidentiality.)

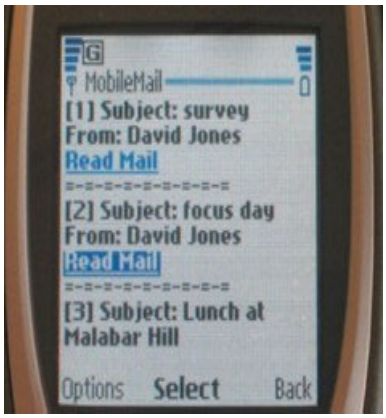

Figure 5. Representative screen for unimodal WAP browser access to email.

An Australian acoustic model was used for speech recognition. Given that a synthetic speech engine with an Australian accent was not available, an engine with an English (UK) accent was used. The GUI for the multimodal system, ePIM, was not unlike the graphical interface available for the unimodal system since it is dependent on what the WAP browser will support (see Figure 6).

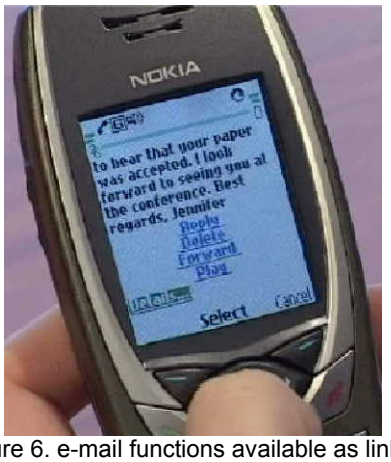

Figure 6. e-mail functions available as links

Experimental Design

A within-subject, repeated measures design was used with each participant using both systems. The order of the mailboxes and the systems was altered to ensure that there would be neither order effect, nor any effect due strictly to the messages in a particular inbox.. The messages in each mailbox were balanced as to length of each message and content, such that the each message in Mailbox 1 had the same word count and was of approximately the same nature as the corresponding message in Mailbox 2. See Table 1 for the comparative readability scores for each mailbox. There were 13 messages in each mailbox.

	Flesch Kincaid (Range 1-10)	Reading Ease 1(hard) - 100(easy)	Reading Grade Level
Mailbox 1	6.2	73.7	7.6
Mailbox 2	6.1	73.2	7.7

Table 1. Flesch readability and ease scores for each mailbox

Participants

Users were invited in for a two hour in-lab study. Participants were 17 adults (10 males and 7 females); employees of three different corporations (that cannot be named at this time), with jobs that were representative of the type of user that would require mobile access to business email. Initially 20 participants were scheduled (balanced for gender), however due to occasional instability problems encountered with the network, only 17 subjects were run. Twelve participants were in the age group 21-35, while nine were in the age group 36-50. To avoid any potential difficulty in recognition or understanding the synthetic speech, all participants were native English-speakers with no reported hearing problems. Participants received a token gift for their participation. All participants signed a consent form and were videotaped.

Tasks

For each system the participant was asked to complete the following tasks.

1. Log on (test account name: user one, password: 111111)
2. Find out how many messages are in your inbox.
3. Find out if you have any messages from Denise Richards, if you do, read the message.
4. Send a reply to that message indicating that you are willing to cover the meeting for that person. However, it is quite possible that your calendar may not be free at that time and you should let the person know that the meeting might have to be scheduled for a different time.
5. Read the 10th message.
6. Forward that message to David Jones, adding the following comment: “Hi David, Please see the attached message for your information.”

Measurements

Both behavioral metrics of participants’ task performance, and subjective measurements of participants’ perception and attitude were measured in the study. After use of each system, the participant completed a questionnaire consisting of attitudinal questions regarding the system,

and their user experience. Participants' demographic information was collected at the end of the questionnaire. All the questions except the demographic ones were measured by asking how well certain adjectives described the system, and how the user felt while using the system on a Likert scale ("0" = "describes very poorly", "7" = "describes very well").

Attitudinal metrics

Three system perception indices were constructed through factor analysis:

1. *Ease of Use*: consisted of "easy to use", "difficult" (reverse coded), and "straightforward"; Cronbach alpha = .823;
2. *Novelty of the system*: consisted of "outdated" (reverse coded), "cutting edge", and "innovative", Cronbach alpha = .824.
3. *Value of the system*: consisted of "useless" (reverse coded), "valuable", and "high quality", Cronbach alpha = .807.

The user experience was measured with the following indices:

1. A composite index of how *draining* the user found the experience was created consisting of "exhausted", "impatient" , and "bored", Cronbach alpha = .836.
2. A correlated index of *engagement* was created consisting of "entertained", "involved" and "interested". Cronbach alpha = .89

Lastly, user satisfaction was measured by having the participants respond to the following questions using a Likert scale after completion of tasks on each system:

1. How satisfied are you overall with the *ease of use* of the system;
2. How satisfied are you overall with the *amount of time* to complete the tasks;
3. How likely would you be to *use this system in the future*;
4. How likely would you be to *recommend* this system to others;

Performance metrics

Participants' performance with each system was measured by collecting the time on task for each task as well as the total time, task completion rates and the number of errors.

FINDINGS

Findings indicate that the multimodal interaction was significantly preferred to the unimodal access for all measurements with the exception of the ease of use index, where the difference was not significant (a preliminary analysis of the attitudinal measures only was reported in a CHI short paper [3]). Time on task for all tasks was significantly faster. A detailed discussion of these results follows.

Attitudal findings

Paired sample T-tests were run to compare the means from the indices collected for each system. The multimodal system was rated higher than the unimodal system for all three system perception indices. Participants' perception of the system's value was significantly lower for the unimodal system (M = 13.81) than the multimodal system (M = 16.93), $t(15) = -2.498$, $\underline{p} < .05$. Participants also thought the multimodal system was more novel (M = 19.50) than the current mobile offering (M = 11.63), $t(15) = -8.08$, $\underline{p} < .001$. Interestingly, the difference in the ease-of-use index, while in the right direction, was not significant, (M = 12.18 for unimodal and M = 13.75 for ePIM), which was perhaps a reflection of the stability problems that we encountered with the 3G network where calls were dropped or would not connect. Figure 7 presents the means for the three indices. (The maximum possible value for each combined index is 21)

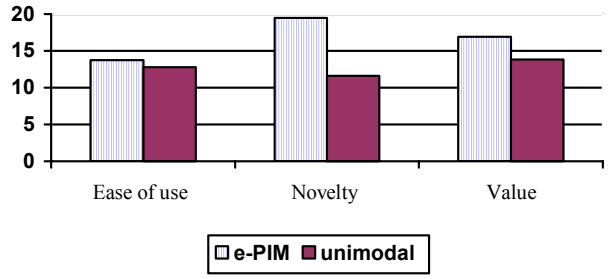

Figure 7. A comparison of means for system perception indices

The user experience indices confirmed the preference for the multimodal system. Participants felt significantly less drained after dealing with the multimodal (M = 7.19) than the unimodal system (M = 11.94) $t(15) = 3.288$, $\underline{p} < .005$. This was most likely due to the need to use multitap keypad input for text creation (email replies or forwarded comments). However, the entire explanation can not be found in the text input requirement given that SMS messaging is quite well established in Sydney and many, if not all, of the participants send and receive SMS messages as part of their daily business and social life, participants had prior experience with inputting text via a telephone keypad. Several participants turned on the predictive T9 dictionary and achieved comfortable input rates. One user was so fluent as to use two-thumb typing on the telephone keypad. Participants also had a significantly higher engagement index with the multimodal system (M = 16.75) than the unimodal system (M = 12.25) $t(15) = -5.411$, $\underline{p} < .001$. Figure 8 charts the means for the user experience indices for both systems (here again the maximum possible value is 21)

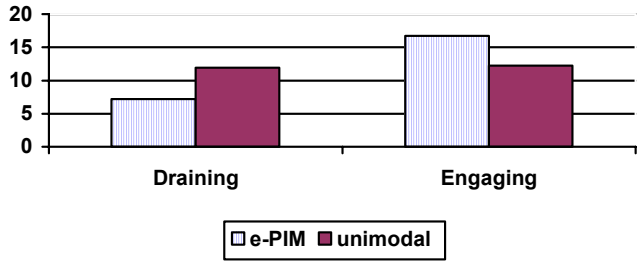

Figure 8. A comparison of means for user experience indices

Additionally, participants indicated that they were significantly more likely to use the ePIM system in the future (M = 6.31) than the unimodal system (M = 3.94) $t(15) = -5.69$, $p < .001$ and significantly more likely to recommend the multimodal system (M = 6.38) than the unimodal system (M = 4.0) $t(15) = -4.76$, $p < .001$. While participants indicated they were significantly more satisfied with the amount of time it took to accomplish the tasks with the multimodal system (M = 4.75) than the unimodal system (M = 2.56) $t(15) = -3.52$, $p < .005$, the difference in their overall satisfaction with the ease of use of the multimodal system compared to the unimodal system, while in the right direction, was not significant. Figure 9 charts the means for all four satisfaction metrics for both systems. (Maximum possible value for each individual index is 7).

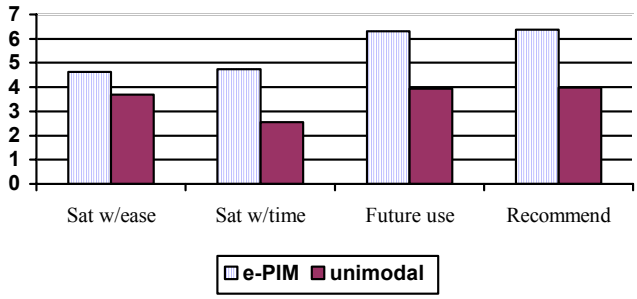

Figure 9. A comparison of means for all four satisfaction metrics

Performance findings

The participants were significantly faster with the multimodal voice system, had very high completion rates on both systems and experienced several usability issues on each system.

The total task time was significantly longer (38%) for the unimodal system (M=12.98 minutes) than for the multimodal system (M=9.49) $t(15) = 2.56$, $p < .05$. If we remove the three participants whose times were increased due to getting stuck in a known ePIM navigation problem, the difference is more pronounced (50%) and still significant (M=13.44 for unimodal and M= 8.96 for ePIM) $t(12) = 2.89$, $p < .05$. If we further remove a participant whose quips and wise cracks (e.g. in response to a yes or no question the participant replied "of course I don't want to delete it you stupid machine") greatly confused the speech recognition engine, the difference in the mean times is greater still (63%) and continues to be significant (M=14.05 for unimodal and M= 8.61 for ePIM) $t(11) = 4.12$, $p < .005$. Figure 10 charts the means for these times (in minutes).

Participants had very high completion rates for both systems. Only one participant did not achieve 100% completion with the multimodal ePIM system (one task was missed) and one participant did not complete a task with the unimodal system.

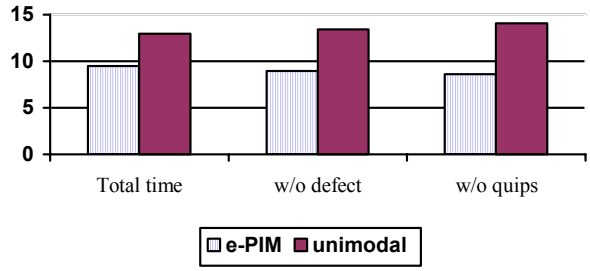

Figure 10. Means for total times in minutes

MODALITY ISSUES

In addition to examining the incremental value of adding speech to a graphical interface in a mobile setting, the user study contributed to increasing the understanding of modality usage (i.e. which modality do users prefer given the task and the circumstances). Our hypothesis was that there would be some distribution of usage between speech and GUI for navigation. However, this was not the case. Speech was unconditionally the dominant modality used in the multimodal system (for both navigation and input). Even people who did not expect to use speech did. One participant, when informed during training that all functions could be accomplished either by speech or by GUI, stated that he intended to only use the GUI, because: *"I'm just more comfortable with graphics"*. However, once the study started, the participant used only speech. When queried about this choice after the fact, he explained: *"I wanted to try using voice because I think that voice is the way that most people would prefer to interact with the system. And what I found is that once you start using it, it is relatively easy to keep going. "*

Certainly part of the explanation for the dominance of the use of the speech modality was the novelty of the interface for these users (note the high ratings for Novelty in Figure 7). While the effect that this had on modality selection is hard to quantify based on a lab study, a longitudinal study (which we hope to accomplish in the third quarter of this year) would help to inform this question. A second explanation is that the users had fairly high recognition

rates. We did not measure speech recognition error rates for each user and task, but recognition rates appeared to be above 90% for most users/tasks. The one exception was when users were instructed to listen to a specific message ("Read the 10th message"). The numeric ("10th") used in the utterance often caused recognition errors. A final explanation for the predominant use of speech in the multimodal system is speed of response time. One participant commented "*It's just easier. It's quicker. And the response time is so quick, it speaks back so quickly. I think if you had to wait a long time for the response that you would probably find it quicker with the keyboard*." As mentioned earlier, it is not clear that this modality dominance would hold up over time. We would expect that as users had time to familiarize themselves with both interfaces they would move fluidly between modalities depending on the circumstances of use.

A further hypothesis of the study was that participants would fall back rather quickly to GUI usage when speech recognition failed. This hypothesis was also based on prior research [9]. Thus we were surprised to find that in most cases users persevered with speech, trying not only alternate phrasings, but returning to phrasings that had previously failed. A representative sequence of utterances when encountering multiple speech recognition errors in a row was:

- Play the 10th message
- Play message number 10
- Read me message number 10
- Play the 10th message

Interestingly, given that the system is a natural language system, all of these utterances are valid and should have worked. Thus it appears there was an issue with the training of the system that caused the problem with the numeric request of a particular message. At any point during these utterances, the user could have scrolled through the list of message headers on the screen until the 10th header was visible, and then selected it through the GUI. A representative comment was "*It's almost like a learning curve thing… I guess that is why I went back to trying different variants. If you can get that (speech) working once as a user, you've got that problem solved forever. It's like an investment in that interface rather than falling back to the screen.*"

USABILITY ISSUES

A usability issue frequently mentioned by participants using ePIM was the problem common to all speech interfaces: not knowing what can be said to the system. When users are presented with a task (e.g. browsing email messages) they want to take an action but are unsure of the words to speak to accomplish this. The power of a natural language system is that there are no "correct" or "incorrect" phrasings, as there are in grammar-based speech systems. However, there are "supported" and "unsupported" functions and users were occasionally unsure whether they had inadvertently wandered into an unsupported function. Thus when a user says "*do I have any messages about the special seminar*" and the system replies: "*I'm sorry I don't understand*" it is unclear to the user whether this is because a key word search is not supported, or that particular utterance was not understood. Since this problem is common to all speech systems, there are a certain number of design techniques that can be applied (given additional development time).

More interestingly, a usability problem that surfaced which is particular to multimodal systems only, is a feeling of "overload" when both modalities are presenting at the same time. As mentioned earlier, when the system presents an email message the text of the message is displayed on the screen, and the text is "spoken" by the system using text-to-speech. Thus if the user had somehow erroneously gotten to this state (either because of a speech recognition error, or a GUI navigation error), and the user is trying to handle getting out of the current condition, the presentation of both modalities was found to be overwhelming.

We had intended to include a mute function to suppress further spoken output from the system but had not had time to implement it before the user trial. We would be sure to include this function in all future versions of the application. The voice system was enabled with barge-in, which allows a user to interrupt the system while it is speaking. Many users intuitively tried this function - sometimes by jokingly telling the system to "shut up", but more often with the term "stop", which worked well. Barge-in however only causes the system to stop speaking its current utterance, and to listen for the next command, usually with an open-end prompt such as "How can I help?" If no reply is received after a while, the system will re-prompt with something along the lines of "I'm sorry, I didn't hear anything." Thus if the user is finding the spoken speech to be distracting, barge-in is not a sufficient remedy.

When the GUI was polling the server it displayed a globe spinning in the upper right-hand corner of the screen (it can be seen in Figure 6). This was our feedback of "GUI busy" to the user, and it indicated that the GUI was not available for interaction. As mentioned earlier, if the user wanted to input text or make a GUI selection while the globe was spinning, he would have to first press Cancel, to interrupt the polling operation. This was explained to users during the training period prior to the start of the study and is reminiscent of the busy indicator available in standard web browsers. However we found that during the study, many users missed this indication and tried to use the GUI even if it was busy. When users fell-back to GUI usage, it was usually because they had encountered multiple problems with speech on a given task, and thus they were probably somewhat flustered at that point. Given that we know multimodal users may have only a divided-attention state to devote to the graphical interface, we should design our

visual feedback to be more apparent.

Additionally, there were many usability issues that participants encountered with the unimodal system. Since the focus of the paper is on the multimodal system, we won't devote much space to these issues but simply mention that text input was a major problem for all participants, even those that were clearly adept at text messaging and using T9 predictive input. Participants also had problems entering the text in the wrong area (e.g., entering the message reply in the area that was dedicated to entering the recipient's address), finding the necessary function (a lot of time was spent looking in vain for Reply in the Options menu) and getting the size and scope of messages in their inbox (for example, they could not tell without scrolling through 4 or 5 screens how many new messages had been received).

CONCLUSION

We have presented e-PIM a conversational multimodal application accepting free-form speech and text input, which was successfully implemented on an unmodified device in a 3G environment. The system was used in a user trial to evaluate its usability and to compare it to a standard unimodal (text only) product for mobile email access. The trial showed that participants significantly preferred the multimodal interaction over the unimodal, were faster, and were much more inclined to use and recommend a multimodal system in the future. We have described the architecture used for the implementation and highlighted issues that can help inform future designers of such systems. More work needs to be done to create a similar system that can accept blended input, which would both give the user more latitude and could possibly create higher recognition rates through use of mutual disambiguation.

ACKNOWLEDGEMENTS

We would like to thank our colleagues at IBM Australia for their invaluable help on making this all happen.

REFERENCES

1. Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E., Sensing Techniques for Mobile Interaction. In *Proceedings of the 13th annual ACM symposium on user interface software and technology, UIST 2000*
2. Jurafsky, D., Martin, J., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, 2000
3. Lai, J., Facilitating Mobile Communication with Multimodal Access to Email Messages on a Cell Phone, In *Proceedings of the Conference on Human Factors in Computing Systems – Short Papers, CHI 2004*
4. Maglio, P., Matlock, T., Campbell, C., Zhai, S., Smith, B., Gaze and speech in attentive user interfaces. In *Proceedings of the Third International Conference on Multimodal Interfaces, ICMI 2000,* Beijing, 2000
5. Neti, C., Iyengar, G., Potamianos, G., Senior, A., Maison, B. Perceptual interfaces for information interaction: Joint processing of audio and visual information for human-computer interaction, In *Proceedings of the International Conference on Spoken Language Processing ICSLP 2000,* vol III, pp. 11-14, Beijing, October 2000.
6. Oviatt, S. Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction (Special Issue on Multimodal Interfaces)* 1997 Volume 12, pp. 93
7. Oviatt, S., Ten myths of multimodal interaction, *Communications of the ACM,* Vol. 42, No. 11, November 1999, pp. 74-81
8. Oviatt, S., Mutual Disambiguation of recognition errors in a multimodal architecture. In *Proceedings of the Conference on Human Factors in Computing Systems CHI '99,* NY, 1999 pp. 576-583
9. Oviatt, S., Multimodal system processing in mobile environments, In *Proceedings of the 13th annual ACM symposium on user interface software and technology, UIST 2000*
10. Oviatt, S. Multimodal interfaces. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, (ed. by J. Jacko and A. Sears), Lawrence Erlbaum Assoc., Mahwah, NJ, 2003, chap.14, 286-304.
11. Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G,, Want, R., TiltType: accelerometer-supported text entry for very small devices. In *Proceedings of the 15th annual ACM symposium on user interface software and technology, UIST 2002*
12. Rekimoto, J., Tilting operations for small screen interfaces. In *Proceedings of the 9th annual ACM symposium on user interface software and technology, UIST 1996*
13. Ruuska, P., Frantti, T., The Multicall Service to Support Multimedia Services in the UMTS Networks. In *Proceedings of the 27th Euromicro Conference*, Warsaw, Poland, September, 2001.
14. Shell, J., Vertegaal, R., Skaburskis, A. EyePliances: attention-seeking devices that respond to visual attention, In *Proceedings of the Conference on Human Factors in Computing Systems – Extended Abstracts, CHI 2003*
15. Vertegaal, R., Dickie, C., Sohn, C., Flickner, M., Designing attentive cell phones using wearable eyecontact sensors, In *Proceedings of the Conference on Human Factors in Computing Systems – Extended Abstracts, CHI 2002*
16. Wigdor, D., Balakrishnan, R., TiltText: using tilt for text input to mobile phones. *Proceedings of the 16th annual ACM symposium on user interface software and technology, UIST 2003*
17. W3C Note 8 January 2003. Multimodal Interaction Requirements http://www.w3.org/TR/mmireqs