Handbook of Research on User Interface Design and Evaluation for Mobile Technology

Volume I

Joanna Lumsden
National Research Council of Canada
Institute for Information Technology – e-Business, Canada

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE

Hershey · New York

Acquisitions Editor: Kristin Klinger
Development Editor: Kristin Roth
Senior Managing Editor: Jennifer Neidig
Managing Editor: Sara Reed
Copy Editor: Joy Langel, Katie Smalley, and Angela Thor
Typesetter: Jeff Ash
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanonline.com

Library of Congress Cataloging-in-Publication Data

Handbook of research on user interface design and evaluation for mobile technology / Joanna Lumsden, editor.

p. cm.

Summary: "This book provides students, researchers, educators, and practitioners with a compendium of research on the key issues surrounding the design and evaluation of mobile user interfaces, such as the physical environment and social context in which a device is being used and the impact of multitasking behavior typically exhibited by mobile-device users"--Provided by publisher.

Includes bibliographical references and index.

ISBN 978-1-59904-871-0 (hardcover) -- ISBN 978-1-59904-872-7 (ebook)

1. Mobile computing--Handbooks, manuals, etc. 2. Human-computer interaction--Handbooks, manuals, etc. 3. User interfaces (Computer systems)--Handbooks, manuals, etc. I. Lumsden, Joanna.

QA76.59.H36 2008

004.165--dc22

2007024493

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter XXVIII
Speech-Centric Multimodal User Interface Design in Mobile Technology

Dong Yu
Microsoft Research, USA

Li Deng
Microsoft Research, USA

ABSTRACT

Multimodal user interface (MUI) allows users to interact with a computer system through multiple human-computer communication channels or modalities. Users have the freedom to choose one or more modalities at the same time. MUI is especially important in mobile devices due to the limited display and keyboard size. In this chapter, we provide a survey of the MUI design in mobile technology with a speech-centric view based on our research and experience in this area (e.g., MapPointS and MiPad). In the context of several carefully chosen case studies, we discuss the main issues related to the speech-centric MUI in mobile devices, current solutions, and future directions.

INTRODUCTION

In recent years, we have seen steady growth in the adoption of mobile devices in people's daily lives as these devices become smaller, cheaper, more powerful, and more energy-efficient. However, mobile devices inevitably have a small display area, a tiny keyboard, a stylus, a low speed (usually less than 400 million instructions per second) central processing unit (CPU), and a small amount (usually less than 64MB) of dynamic random-access memory. Added to these limitations is the fact that mobile devices are often used in many different environments, such as dark and/or noisy surroundings, private offices, and meeting rooms. On these devices, the traditional *graphical user*

interface (GUI)-centric design becomes far less effective than desired. More efficient and easy-to-use user interfaces are in urgent need. The *multimodal user interface* (MUI), which allows users to interact with a computer system through multiple channels such as speech, pen, display, and keyboard, is a promising user interface in mobile devices.

Multimodal interaction is widely observed in human-human communications where senses such as sight, sound, touch, smell, and taste are used. The research on multimodal human-computer interaction, however, became active only after Bolt (1980) proposed his original concept of "Put That There." Since then, a great amount of research has been carried out in this area (Bregler, Manke, Hild, & Waibel 1993; Codella, Jalili, Koved, Lewis, Ling, Lipscomb, et al., 1992; Cohen, Dalrymple, Moran, Pereira, Sullivan, Gargan, et al., 1989; Cohen, Johnston, McGee, Oviatt, Pittman, Smith, et al., 1997; Deng & Yu, 2005; Fukumoto, Suenga, & Mase, 1994; Hsu, Mahajan, & Acero 2005; Huang, Acero, Chelba, Deng, Droppo, Duchene, et al., 2001; Neal & Shapiro, 1991; Pavlovic, Berry, & Huang, 1997; Pavlovic & Huang, 1998; Vo, Houghton, Yang, Bub, Meier, Waibel, et al., 1995; Vo & Wood, 1996; Wang, 1995). Importantly, the body of this research work pointed out that MUIs can support flexible, efficient, and powerful human-computer interaction.

With an MUI, users can communicate with a system through many different input devices such as keyboard, stylus, and microphone, and output devices such as graphical display and speakers. MUI is superior to any single modality where users can communicate with a system through only one channel. Note that using an MUI does not mean users need to communicate with the system always through multiple communication channels simultaneously. Instead, it means that users have freedom to choose one or several modalities when communicating with the system, and they can switch modalities at any time without interrupting the interaction. These characteristics make the MUI easier to learn and use, and is preferred by users in many applications that we will describe later in this chapter.

MUI is especially effective and important in mobile devices for several reasons. First, each modality has its strengths and weaknesses. For this reason, single modality does not permit the user to interact with the system effectively across all tasks and environments. For example, speech UI provides a hands-free, eyes-free, and efficient way for users to input descriptive information or to issue commands. This is very valuable when in motion or in natural field settings. Nevertheless, the performance of speech UI decreases dramatically under noisy conditions. In addition, speech UI is not suitable when privacy and social condition (e.g., in a meeting) is a concern. Pen input, on the other hand, allows users to interact with the system silently, and is acceptable in public settings and under extreme noise (Gong, 1995; Holzman, 1999). Pen input is also the preferred way for entering digits, gestures, abbreviations, symbols, signatures, and graphic content (Oviatt & Olsen, 1994; Suhm, 1998). However, it is impossible for the user to use pen input if he/she is handicapped or under "temporary disability" (e.g., when driving). MUI, on the other hand, allows users to shift between modalities as environmental conditions change (Holzman, 1999), and hence, can cover a wider range of changing environments than single-modal user interfaces.

Second, different modalities can compensate for each other's limitations and thus provide users with more desirable experience (Deng & Yu, 2005; Oviatt, Bernard, & Levow, 1999; Oviatt & vanGent, 1996; Suhm, 1998). For example, the accuracy of a resource-constrained, midsized vocabulary speech recognizer is low given the current speech technology. However, if the speech recognizer is used together with a predictive T9 (text on 9 keys) keyboard, users can greatly increase the text input throughput compared with using the speech modality or T9 keyboard alone (Hsu et al., 2005). The gain is obtained from the mutual disambiguation effect, where each error-prone modality provides partial information to aid in the interpretation of other modalities. Another reason for the improved user experience is users' active error avoidance, where users tend to select the input modality that they judge to be less error

prone for a particular task and environment (Oviatt & vanGent, 1996), and tend to switch modalities to recover from system errors (Oviatt et al., 1999). Mutual compensation is very important for mobile devices because the ability of every single modality in the devices is extremely limited (e.g., a limited display and keyboard size, and limited speech recognition accuracy).

Despite the importance of MUI in mobile devices, designing effective MUIs is far from trivial. Many MUIs in mobile devices are *speech centric*, where speech is the central and main modality. In this chapter, we will focus on main issues on the design of effective speech centric MUIs in mobile devices based on our research and experience in developing MapPointS (Deng & Yu, 2005) and MiPad (Deng, Wang, Acero, Hon, Droppo, Boulis, et al., 2002; Huang, Acero, Chelba, Deng, Droppo, Duchene, et al., 2001). In Section 2, we describe a generic MUI architecture in mobile setting that consists of various recognizers for different input modalities, semantic parsers, a discourse manager, and a response manager. In Section 3, we discuss special considerations related to speech modality. In particular, we discuss the approaches to overcoming resource limitations on mobile devices, noise robust speech front-ends, noise robust modality switching interfaces, and context-aware language model. In section 4, we introduce the issues related to robust natural language understanding including construction of robust grammars. We discuss the problem of modality fusion, including modality-neutral semantic representation, unification approach, and modality integration, in Section 5. We discuss possible future directions and conclude this chapter in Section 6.

A GENERIC MUI ARCHITECTURE

The ultimate goal of an MUI is to fulfill the needs and requirements of the users. This principle is one of many emphasized in *user-centered design* (Gould & Lewis, 1985, Norman & Draper, 1986). According to the user-centered design principle, the acceptability of an MUI can be judged using three main attributes (Dybkjaer & Bernsen, 2001; Hone & Graham, 2001; Nielsen, 1993): effectiveness, efficiency, and learnability. The *effectiveness* assesses whether users can complete the tasks and achieve the goals with the predefined degree of perceived accuracy. It is usually measured on the targeted user population, over a specified range of tasks and environments. The *efficiency* judges how much effort (cognitive demand, fatigue, stress, frustration, discomfort, and so on) and resources (time) are needed for users to perform specific tasks. It is usually measured with the total time (including time for error corrections) taken to complete a task. The *learnability* measures whether users can easily discover the system's functionality and quickly learn to use the system.

Figure 1 depicts a typical speech-centric MUI architecture that is aimed to achieve a high level of effectiveness, efficiency, and learnability. As shown in the figure, users can communicate with the system through speech, keyboard, and other modalities such as pen and camera. Modality fusion usually is the center of an MUI system. There are two typical ways of fusing information from different input modalities, namely, early fusion and late fusion. With the *early fusion*, signals are integrated at the *feature* level and hence, the recognition process in one modality would affect that in another modality (Bregler et al., 1993, Pavlovic et al., 1997; Pavlovic & Huang, 1998; Vo et al., 1995,). Early fusion is suitable for highly coupled modalities such as speech and lip movements (Rubin, Vatikiotis-Bateson, & Benoit, 1998; Stork & Hennecke, 1995). However, early fusion can greatly increase the modeling complexity and computational intensity due to its nature of intermodality influence in the recognition phase. With the *late fusion*, information is integrated at the *semantic* level. The benefit of late fusion is its isolation of input modalities from the rest of the system. In other words, individual recognizers trained using unimodal data can be directly plugged into the system without affecting the rest of the system. This feature makes the late fusion easier to scale up to more modalities in the future than the early fusion. The architecture shown in Figure 1 utilizes the late fusion approach that has

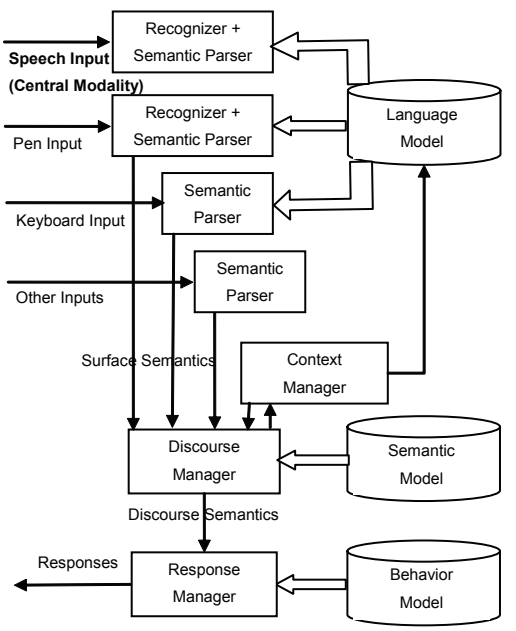
been widely adopted, for example, by a variety of systems including Put-That-There (Bolt, 1980), MapPointS (Deng & Yu, 2005), MiPad (Huang et al., 2001), ShopTalk (Cohen, et al., 1989), QuickSet (Cohen, Johnston, McGee, Oviatt, Pittman, Smith, et al., 1997), CUBRICON (Neal & Shapiro, 1991), Virtual World (Codella, Jalili, Koved, Lewis, Ling, Lipscomb, et al., 1992), Finger-Pointer (Fukumoto et al., 1994), VisualMan (Wang, 1995), and Jeanie (Vo & Wood, 1996).

In the late-fusion approach depicted in Figure 1, the input signals received by the system are first processed by *semantic parsers* associated with the corresponding modality into the *surface semantics* representation. Note that although each modality has its own semantic parser, the resulting surface semantics are represented in a common semantic representation and is thus independent of the modality. The surface semantics from all the input modalities are then fused by the *discourse manager* component into the *discourse semantics* representation (more discussions on this issue in Section 4). In order to generate discourse semantics, the discourse manager uses the semantic modal and interacts with the context manager to utilize and update such information as dialog context, domain knowledge, user's information, and user's usage history. The updated context information can be used to adapt the language model, which can improve speech recognition accuracy and enhance the quality of semantic parsers for the next user-computer interaction.

The discourse semantics, which is the output of the discourse manager, is then fed into the *response manager* to communicate back to the user. The response manager synthesizes the proper responses, based on the discourse semantics and the capabilities of the user interface, and plays the response back to the user. In this process, behavior model provides rules to carry out the required actions. The combination of discourse manager and response manager is usually referred to as the dialog manager.

Note that the components shown in Figure 1 may reside on the mobile devices, or distributed on other servers in real implementations. In addition, many MUI systems use an agent-based software solution in which a facility or hub is used to pass information to and from different components (or agents) (Kumar & Cohen, 2000; Schwartz, 1993).

Figure 1. A typical speech-centric MUI architecture and its components



Many best practices and design principles have been developed for the speech-centric MUI design in the past decades (Becker, 2001; Dybkjaer & Bernsen, 2001; Ravden & Johnson, 1989; Reeves, Lai, J., Larson, J.A., Oviatt, S., Balaji, T.S., Buisine, et al. 2004), which we summarize next.

First, the system should explicitly inform the user about its state through appropriate feedback within a reasonable amount of time, so as to avoid state errors, that is, the user's perceived state is different from the system's perceived state. The feedback can be in different modalities, but must be clear and accurate. If speech feedback is used, recorded speech is usually preferred over the synthesized speech, due to its higher degree of naturalness. Note that the recorded speech usually takes a larger amount of resources than the synthesized speech. Since the memory and storage available in mobile devices is very limited, designers should strike a balance between the use of synthesized speech and of recorded speech. The system should follow real-world conventions, and use the words, phrases, and concepts that are familiar to the users. The system should also ensure that the output modalities be well synchronized temporally. For example, the spoken directions should be synchronized with the map display.

Second, the system should provide sufficient flexibility so that users can select the modalities that are best for the task under the specific environments. For example, the user should be able to switch to a nonspeech modality when inputting sensitive information such as personal identification numbers and passwords. A good MUI design should also allow users to exit from an unwanted state via commands that are global to the system, instead of having to go through an extended dialog. The system should provide enough information (e.g., through prompts) to guide novice users to use the system, yet at the same time allow barge-ins and accelerators for the expert users to reduce the overall task completion time.

Third, the system should be designed to allow easy correction of errors. For example, the system should provide context sensitive, concise, and effective help. Other approaches include integrating complementary modalities to improve overall robustness during multimodal fusion; allowing users to select a less error-prone modality for a given lexical content, permitting users to switch to a different modality when error happens; and incorporating modalities capable of conveying rich semantic information.

Fourth, the system's behavior should be consistent internally and with users' previous experiences. For example, a similar dialog flow should be followed and the same terms should be used to fulfill the same task. Users should not have to wonder whether the same words and actions have different meaning under different context.

Fifth, the system should not present more information than necessary. For example, dialogues should not contain irrelevant or rarely needed information, and the prompts should be concise.

While the best practices summarized are common to all speech-centric MUIs, some special attention needs to be paid to speech modality and multimodality fusion due to the great variations of mobile device usage environments. We address these special considerations next.

SPECIAL CONSIDERATIONS FOR SPEECH MODALITY

There are two main challenges for the use of speech modality on mobile devices. First, the resources on mobile devices, in particular, CPU speed, memory, and communication bandwidth, are very limited. Second, speech recognition accuracy degrades substantially in realistic noisy environments, where there are abrupt changes in noise, or variable phase-in phase-out sources of noise as the user moves. For example, the recognition accuracy may drop 30-50% inside a vehicle and cafeteria from that in a quiet environment (Das, Bakis, Nadas, Nahamoo, & Picheny, 1993; Lockwood & Boudy, 1992). Since the mobile devices will be used in these real-field settings without a close-talk microphone, robustness to acoustic environment, that is, immunity to noise and channel distortion, is one of the most important aspects to consider when designing speech-centric MUIs on mobile devices. Speech

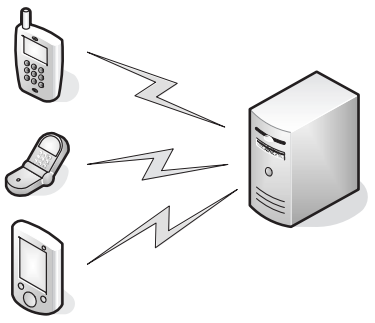
recognition accuracy and robustness can usually be improved with a noise-robust speech front-end, a noise-robust modality-switching interface, and a context aware language model.

Resource Constrained Speech Recognition

Speech recognition on mobile devices is typically carried out with two options: the *distributed recognition* (Deng et al., 2002) where the recognition happens at a remote server (Figure 2) and the *local recognition* (Deligne, Dharanipragada, Gopinath, Maison, Olsen, & Printz, 2002; Varga, Aalburg, Andrassy, Astrov, Bauer, Beaugeant, et al., 2002) where the recognition is carried out completely on the mobile device. The distributed recognition can take advantage of the power of the remote server to achieve a fast and accurate recognition, while the local recognition can eliminate the requirement of the device to have a fast data connection.

In the distributed architecture, the main consideration is the latency required to send data to and from the server. The latency is typically determined by the communication bandwidth and the amount of data sent. To reduce the latency, a typical approach is to use a standard codec on the device to transmit the speech to the server where the coded speech is subsequently decompressed and recognized (as depicted in Figure 3). However, since speech recognizers only need some features of the speech signal (e.g., Mel-cepstrum), an alternative approach is to put the speech front end on the mobile device and transmit only speech features to the server (Deng et al. 2002), as shown in Figure 4. Transmitting speech features can further save bandwidth because the size of the features is typically much less than that of the compressed audio signals.

Figure 2. Illustration of distributed speech recognition where the actual recognition happens at the server (e.g., PC)



Besides the advantage of using the computing power at the server to improve speech recognition accuracy, there are other benefits of using server-side recognition. One such benefit is its better maintainability compared to the local recognition approach because updating software on the server is much easier and more cost effective than updating software on millions of mobile devices. It, however, does require the recognizer on the server to be front end or codec agnostic in order to materialize this benefit. In other words, the recognizer should make no assumptions on the structure and processing of the front end (Deng et al., 2002). Another benefit of using distributed recognition is the possibility for the server to personalize the acoustic model, language model, and understanding model all at the server, saving the precious CPU and memory on mobile devices. In the past, distributed recognition is unquestionably the dominant approach due to the low CPU speed and small amount of memory available on the mobile devices. Nowadays, although the CPU speed and memory size are increasing dramatically, distributed recognition is still the prevailing approach over local recognition due to the advantages discussed previously.

The major issue of the local recognition architecture is the low recognition speed and accuracy due to the slow CPU speed and low memory available on mobile devices. Speech recognizers running on mobile devices need to be specially designed (Deligne et al., 2002, Li, Malkin, & Bilmes, 2006; Varga, Aalburg, Andrassy, Astrov, Bauer, Beaugeant, 2002) to fit the requirement since speech recognizers designed for the desktop or telephony systems cannot be directly deployed to mobile devices. The greatest benefit of using the local recognition approach is its independency of the network connection and the server and

Figure 3. Distributed speech recognition architecture: speech input is encoded and sent to the server. Speech feature extraction happens at the server side

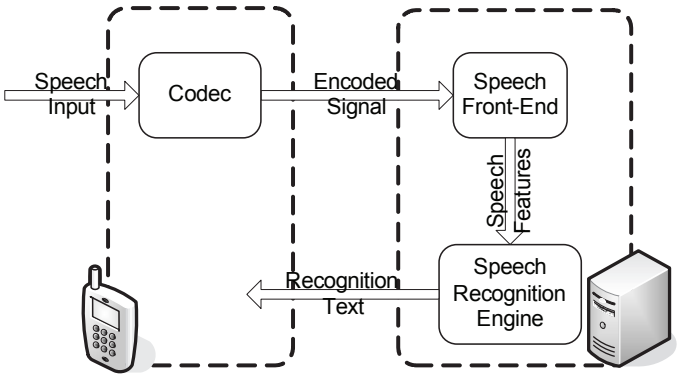
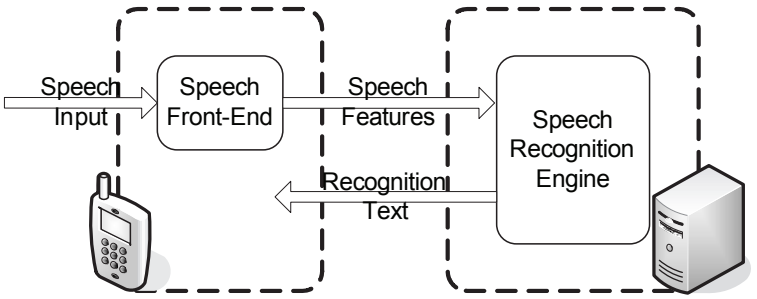


Figure 4. Distributed speech recognition architecture alternative: the speech feature extraction happens on the mobile devices. Only the features are sent to the server



hence, can be used everywhere under any conditions. Given the consistent improvement of the CPU speed and memory on the mobile device hardware, in the future, the local recognition approach is expected to become more and more popular for simple tasks such as name dialing and media playing.

Noise Robust Speech Front End

Noise robustness is one of the most important requirements for speech-centric MUI on mobile devices. It has attracted substantial attention in the past several years. Many algorithms have been proposed to deal with nonstationary noises. A popular one is an advanced feature extraction algorithm (jointly developed by Motorola Labs, France Telecom and Alcatel) that was selected in February of 2002 as a standard in distributed speech recognition by the European telecommunications standards institute. The algorithm defines the extraction and compression of the features from speech that is performed on a local, terminal device, for example, a mobile phone. These features are then sent over a data link to a remote "back-end processor" that recognizes the words spoken. The major components of this algorithm are noise reduction, waveform processing, cepstrum calculation, blind equalization, and voice-activity detection. The noise reduction component makes use of two-stage Wiener filtering (Macho, Mauuary, Noé, Cheng, Ealey, Jouvet, et al., 2002).

The stereo-based piecewise linear compensation for environments (SPLICE), which has been used in the MiPad system (Deng et al., 2002), is another effective algorithm for noise robust speech feature extraction. SPLICE is a cepstrum enhancement algorithm dealing with additive noise, channel distortion, or a combination of the two. It is a dynamic, frame-based, bias-removal algorithm with no explicit assumptions made on the nature of the noise model. In SPLICE, the noise characteristics are embedded in the piecewise linear mapping between the "stereo'' clean and distorted speech cepstral vectors. SPLICE has a potential to handle a wide range of distortions, including nonstationary distortion, joint additive and convolutional distortion, and nonlinear distortion (in time-domain), because SPLICE can accurately estimate the correction vectors without the need for an explicit noise model.

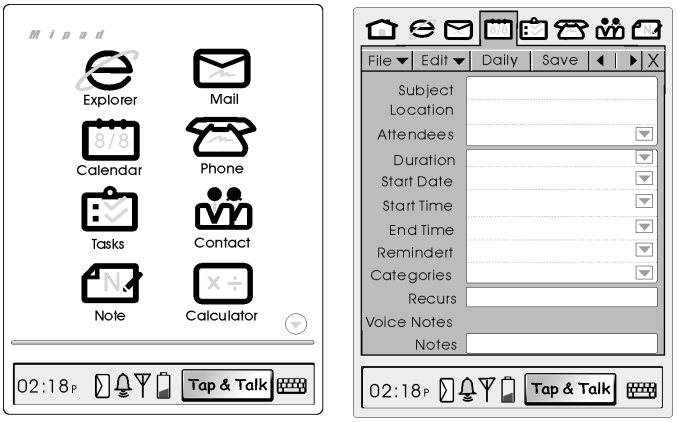
Modality Switching

One of the problems in speech recognition under noisy environment is modality switching. If the speech recognition engine is always on, noises and by-talks may be misrecognized as a legitimate user input and hence, can erroneously trigger commands.

A widely used modality switching approach is called "push to talk," where the user presses a button to turn on the speech recognizer, and releases the button to turn off the recognizer. Another approach is called "tap & talk" (Deng et al., 2002; Huang, Acero, A., Chelba, C., Deng, L., Duchene, D., Goodman, et al., 2000, Huang et al., 2001), where the user provides inputs by tapping the "tap & talk" field and then talking to it. Alternatively, the user can select the tap & talk field by using the roller to navigate and holding it down while speaking. Tap & talk can be considered as a combination of push-to-talk control and indication of where the recognized text should go. Both the push-to-talk and tap & talk avoid the speech detection problem that is critical to the noisy environment under which the mobile devices are typically deployed.

Figure 5 shows an example of the tap & talk interface used in the MiPad (Deng et al., 2002). If the user wants to provide the attendee information for a meeting scheduling task, he/she taps the "attendees" field in the calendar card. When that happens, the MUI will constrain both the language model and the semantic model based on the information on the potential attendees. This can significantly improve the accuracy and the throughput. Note that tap & talk functions as a user-initiative dialog-state specification. With tap & talk, there is no need for the mobile devices to include any special mechanism to handle spoken dialog focus and digression.

Figure 5. An example of the Tap & Talk interface (Deng et al., 2002, © 2002 IEEE)



Context-Aware Language Model

Here, *context* refers to any information that can be used to characterize the situation related to human-computer interaction. It typically includes the surrounding environment (e.g., location and noise condition), the user (e.g., age and gender, preferences, past interaction experiences, and the interaction history in the current session), and the devices (e.g., remaining battery life, available memory, screen-size, screen-contrast, and speaker volume). Although context-awareness can be beneficial to all components in an MUI, it is especially important for improving speech recognition accuracy under noisy environments.

Context information can be utilized in many different ways in speech modality. One particular approach is to construct the language model based on the context. For example, the tap & talk approach (Deng et al., 2002) customizes the language model depending on the field the user is pointing to, as mentioned in section 3.3.

Language model can also be customized, based on the user information and the dialog state. For example, if the system is expecting the recipient information, the language model can include only the names in the global address book. If the user information is also used, the language model can also include user's contact list and people who have exchanged e-mails with the user in the past. An even more effective language model would weight different names differently, depending on the frequencies the user exchanged e-mail with the person, and the recentness of the interaction (Yu, Wang, Mahajan, Mau, & Acero, 2003). Another example of constructing the language model based on the context and user information is described in the speech enabled MapPoint (Deng & Yu, 2005). Without context information, the speech recognizer needs to load all location names and business names in the North America. This is definitely beyond the ability of most state-of-the-art speech recognizers. However, if the user's location information and/or the interaction history are known, the system can load only the location names and business names around the user's current location, and weight all the names based on the popularity of the names as well as the user's interaction history.

A more advanced context-aware language model construction technique is discussed by Wang (2004). This detection-based technique is used in the second generation of the MiPad (Wang, 2004). The basic idea of this approach is to detect the context cues from the user's partial utterances sequentially, and adjust the language model dynamically for the next part of the utterances. This approach has achieved excellent user experience.

LANGUAGE UNDERSTANDING

Good speech recognition accuracy does not always translate to good understanding of users' intents, as indicated by Wang, Acero, and Chelba (2003). A robust language-understanding model is needed to obtain good user experience for speech-centric MUI applications, especially since speech recognition errors will affect the understanding.

The first issue to address in language understanding is constructing the semantic grammar. Since the importance of each word to the understanding is different, the words need to be treated differently. A typical approach is to introduce a specific type of nonterminals called semantic classes to describe the schema of an application (Wang, 2001; Yu, Ju, Wang, & Acero, 2006). The semantic classes define the concepts embedded in the linguistic structures, which are usually modeled with probabilistic context-free grammars. The advantage of introducing the semantic classes is to make the linguistic realization of semantic concepts independent of the semantic concepts themselves. Once the semantic classes are defined, a robust linguistic grammar can be built using the approaches similar to the one described by Yu, et al. (2006).

The transformation from the recognized text to the semantic representation is usually done using a semantic parser. For example, in MiPad, this transformation is done using a robust chart parser (Wang, 2001). In this parser, "the robustness to ungrammaticality and noise can be attributed

to its ability of skipping minimum unparsable segments in the input. The algorithm uses dotted rules, which are standard context free grammar rules in Backus Naur form plus a dot in front of a right-hand-side symbol. The dot separates the symbols that already have matched with the input words from the symbols that are yet to be matched." (Wang, 2001, pp. 1556) Since the language models used in MiPad are dynamically generated based on the current user information and the tap & talk field, the parser used in MiPad supports dynamic grammars. Given that some part of the user's utterances is in the free-style form (e.g., the topic of a meeting to be scheduled), they are modeled as dictation grammar rules. Since speech recognition is not perfect, the MiPad robust parser takes into account the N-best list, together with the associated confidence scores returned from the speech recognition engine, and combines the speech recognition score with the parsing score to obtain the best parsing result. More recent progress includes using maximum entropy models to classify the tasks and to disambiguate the meaning of the slots in the recognition result.

MODALITY FUSION

One strong advantage of using MUIs is the improved accuracy and throughput through modality integration. There are typically two fusion approaches: early fusion and late fusion. Given that late fusion has many superior properties over the early one, as discussed in Section 2, it will be the focus of our discussion in this section. There are two tasks in the late fusion: Process and convert the input signals into a common surface semantic representation using the semantic parsers (one specific to each modality), and fuse the surface semantics into discourse semantics using the discourse manager.

Semantic Representation and Unification

The semantic fusion operation requires a meaning representation framework that is common among modalities, and a well-defined operation for combining partial meanings.

Many semantic representation formats have been proposed in the past. For example, in Bolt's (1980) pioneering paper, only very limited modality fusion is required and hence, a simple semantic representation was used. In the past decade, researchers (Cheyer & Julia, 1995; Pavlovic & Huang, 1998; Shaikh, Juth, Medl, Marsic, Kulikowski, & Flanagan, 1997; Vo & Wood, 1996) have converged to using a data structure called typed *feature structures* (Kay, 1979) to represent meanings. Typed feature structure can be considered as an extended, recursive version of attribute-value-type data structures, where a value can, in turn, be a feature structure. It extends *frames* (Minsky, 1975) that represent objects and relations as nested sets of attribute/value pairs, by using shared variables to indicate common substructures. A typed feature structure indicates the kind of entity it represents with a type, and the values with an associated collection of feature-value or attribute-value pairs. In the typed feature structure, a value may be nil, a variable, an atom, or another typed-feature structure.

The primary operation on typed feature structure is *unification*. "*Typed-feature-structure unification* is an operation that determines the consistency of two representational structures and, if they are consistent, combines them into a single result." (Oviatt, Cohen, Wu, Vergo, Duncan, Suhm, et. al., 2000, online version pp. 21) Unification can combine complementary input from different modalities and rule out contradictory input (Johnston, 1998).

Note that users' multimodal inputs may involve *sequentially integrated* or *simultaneously delivered* signal fragments. In other words, temporal relationships between different input channels are very important. To fuse modalities, we need to first determine whether two input fragments are related. In most of the systems reported, this is achieved by considering all input contents that lie within a predefined time window. To do this, all input fragments need to be time stamped as soon as they are generated to remove the errors due to transit delays.

For example, the speech input "Show me the restaurants around here." might have a gesture-input accompanying it either "before," "during," or "after" the actual utterance, and all these three possibilities should provide the same result. Usually the term "before" represents a timeframe of up to several minutes, "during" represents a timeframe of 4 to 5 seconds, and "after" represents a timeframe of 500ms to 750ms. If these values are too small, many multimodal inputs will be considered as unimodal inputs and will not be integrated. If the values are too large the chances of an old or invalid user input are likely being accepted as part of a valid multimodal input.

To determine whether two input fragments should be treated as parts of a multimodal construction or separate unimodal commands, knowledge gained from a user study is very helpful. For example, it has been shown in Oviatt, DeAngeli, and Kuhn (1997) that users' written input precedes speech during a sequentially integrated multimodal command. They have also clarified the distribution of typical intermodal lags.

Semantic Fusion with Uncertain Inputs

The challenge of semantic fusion with uncertain inputs is to determine the unified meaning based on multimodal input fragments associated with probabilities. This is especially important for speech-centric MUI because the output of a speech recognizer is never certain. Note that the unification operation on the typed feature structure assumes that all input modalities are certain, and so they cannot be directly applied here. To fuse modalities with uncertainties, a *hybrid symbolic/statistical* architecture that combines statistical processing techniques with a symbolic unification-based approach is in need. This combined approach involves many factors when fusing the semantics. These factors include recognition accuracy of the individual modalities, the way of combining posterior probabilities, and the prior distribution of multimodal commands.

Note that a multimodal input gives rise to three different types of information overlay: nonoverlayed, overlayed and nonconflicting, and overlayed and conflicting. Nonoverlayed information indicates that the input (unimodal or multimodal) does not have any of the same information represented multiple times. This is the simplest condition. Overlayed and nonconflicting information refers to information segments that may have been represented multiple times without a conflict. The overlayed and conflicting information refers to the case that the information has been provided multiple times and conflicts. There are many approaches to resolving conflicting information in typed feature structure if no uncertainty is involved. The "unification" approach simply returns the value null when a conflict is detected. The "overlay" method returns the first argument when conflicting information is present. However, given that the semantic information from different modalities should not be equally trusted, a better conflicting information resolving approach can be found to handle input signals that may or may not be overlapped in their temporal delivery (Oviatt et al., 1997). Note that overlayed information may arise when inputs are from different modalities (e.g., speech and gesture), or when the same-type modality information occurs multiple times over an extended time frame. Both these two conditions need to be handled.

Conventionally, the probability of the merged feature structures is the cross product of the probabilities of individual feature structures based on the assumption that inputs are statistically independent with each other. In this section, we describe an alternative statistical approach that has been used in QuickSet (Wu, Oviatt, & Cohen, 1999). This approach uses the associative map to reduce the unification pairs and members-teams-committee (MTC) model to refine the multimodal integration process so that different weights are assigned to different modes and different constituents.

Associative map defines all semantically meaningful mapping relations that exist between different sets of constituents for each multimodal command. In its simplest form, it can be considered as a simple process of table lookup. For example, if an MUI consists of only the speech modality and the pen modality, we can build a two-dimensional

table. If two inputs from different modalities can be fused, the value at the corresponding cell is 1; otherwise, the value is 0. The purpose of the associative map is to rule out considerations of those feature structures that cannot possibly be unified semantically.

Members-teams-committee weighs the contributions derived from different modality recognizers based on their empirically-derived relative reliabilities. MTC consists of multiple members, multiple teams, and a committee. "*members* are the individual recognizers that provide a diverse spectrum of recognition results (local posterior probabilities). Member recognizers can be on more than one team. Members report their results to their recognizer *team* leader, which then applies various weighting parameters to their reported scores. Furthermore, each team can apply a different weighting scheme, and can examine different subsets of data. Finally, the *committee* weights the results of the various teams, and reports the final recognition results. The parameters at each level of the hierarchy are trained from a labeled corpus." (Oviatt, et al., 2000, online version, p. 24).

CONCLUSION AND FUTURE DIRECTIONS

In this chapter, we discussed the importance of using the MUI in mobile devices, and described the state-of-the-art technologies in designing speech-centric MUI in mobile devices. Specifically, we discussed the noise robustness technologies, the reliable modality switching methods, the context-aware language model, and the robust language-understanding technologies that contribute to the usability of the speech modality. We also described the modality integration technologies that are important to improving the accuracy and throughput of the MUI. Although these technologies have greatly advanced the speech centric MUI design and development in the mobile devices, future research is needed in the following areas.

Microphone Array Processing

Noise robustness is still a challenging research area for speech-centric MUIs. Although many single-microphone noise robustness technologies (e.g., Deng, et al., 2002; Macho, et al. 2002) have been proposed to improve speech recognition accuracy under noisy environments, the progress so far is still limited. Given the continuous decrease in the hardware price, using microphone array on mobile devices is a trend to combat noisy acoustic conditions and to further decrease speech recognition errors. Microphone array algorithms, which take advantage of the received signal differences between microphones, can achieve noise suppression of 10-15 db effectively (Tashev & Malvar, 2005). Future research is needed for more efficient and effective algorithms using low-cost, low-quality microphone arrays that may be equipped in speech-centric mobile devices.

Error Handling Techniques

Fragile error handling continues to be a top interface problem for speech-centric MUI (Karat, Halverson, Horn, & Karat, 1999; Rhyne & Wolf, 1993; Roe & Wilpon, 1994). A great amount of research work needs to be done in developing graceful error-handling strategies in speech-centric MUI. First, new statistical methods need to be developed to reduce errors through mutual disambiguation between modalities. Second, new dialog strategies (e.g., mixed initiative) need to be developed to allow easy correction of the errors. Third, the system needs to be able to adapt to different environments and challenging contexts to reduce errors. Fourth, better robust speech recognition technologies need to be developed to increase the speech recognition accuracy under a wide range of environments.

Adaptive Multimodal Architectures

In most current MUI systems, their behaviors are predesigned by the developers. The system does not

automatically learn to improve the performance as users use the system. Given that mobile devices are usually used by a single user, it is very important to develop adaptive MUI architectures.

For example, Oviatt (1999) showed that any given user's habitual integration pattern (simultaneous vs. sequential) is apparent at the beginning of their system interaction. When the user uses the system, the interaction pattern remains the same. An adaptive MUI system that can distinguish and utilize these patterns to improve the modality fusion could potentially achieve greater recognition accuracy and interactive speed. Another example is for the system to gradually change the behavior (e.g., automatically predict the user's next action) when the user changes from a novice to an experienced user.

Future research in this area would include what and when to adapt, as well as how (e.g., through reinforcement learning) to adapt MUI systems so that their robustness can be enhanced.

Mixed Initiative Multimodal Dialog

Most current speech-centric MUI systems are user initiative, where the user controls the dialog flow (for example, through push to talk). A user-initiative system can be modeled as a set of asynchronous event handlers. In a more advanced system, the system should also actively interact with the user to ask for missing information (which is called mixed initiative). For example, if the user wants to search for the phone number of a business using a mobile device and he/she forgets to mention the city and state information, the dialog system should automatically ask the user for that information through the multimodal output devices.

Future research should address the design and development of consistent and efficient conversational interaction strategies that can be used by different multimodal systems. Multimodal dialogue systems should be developed within a statistical framework (Horvitz, 1999) that permits probabilistic reasoning about the task, the context, and typical user intentions.

REFERENCES

Becker, N. (2001). *Multimodal interface for mobile clients*, Retrieved July 16, 2006, from http://citeseer.ifi.unizh.ch/563751.html

Bolt, R. A. (1980). Put-that-there: Voice and gesture at the graphics interface. *Computer Graphics, 14(3),* 262-270.

Bregler, C., Manke, S., Hild, H., & Waibel, A. (1993). Improving connected letter recognition by lipreading. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 1*, 557-560.

Cheyer, A., & Julia, L. (1995). Multimodal maps: An agent-based approach. *International Conference on Cooperative Multimodal Communication* (pp. 103-113).

Codella, C., Jalili, R., Koved, L., Lewis, J., Ling, D., Lipscomb, J., Rabenhorst, D., Wang, C., Norton, A., Sweeney, P., & Turk, C. (1992). Interactive simulation in a multi-person virtual world. *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 329-334).

Cohen, P. R., Dalrymple, M., Moran, D. B., Pereira, F. C. N., Sullivan, J. W., Gargan, R. A., Schlossberg, J. L., & Tyler, S. W. (1989). Synergistic use of direct manipulation and natural language. *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 227-234).

Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., & Clow, J. (1997). Quickset: Multimodal interaction for distributed applications. *Proceedings of the Fifth ACM International Multimedia Conference* (pp. 31-40).

Das, S., Bakis, R., Nadas, A., Nahamoo, D. & Picheny, M. (1993). Influence of background noise and microphone on the performance of the IBM TANGORA speech recognition system. *Proceedings of the IEEE International Conference on Acoustic Speech Signal Processing* (pp. 71-74).

Deligne, S., Dharanipragada, S., Gopinath, R., Maison, B., Olsen, P., & Printz, H. (2002). A ro-

bust high accuracy speech recognition system for mobile applications. *IEEE Transactions on Speech and Audio Processing*, *10*(8), 551-561.

Deng, L., Wang, K., Acero, A., Hon, H., Droppo, J., Boulis, C., Wang, Y., Jacoby, D., Mahajan, M., Chelba, C., & Huang, X.D. (2002). Distributed speech processing in MiPad's multimodal user interface. *IEEE Transactions on Speech and Audio Processing*, *10*(8), 605-619.

Deng, L, & Yu, D. (2005). A speech-centric perspective for human-computer interface - A case study. *Journal of VLSI Signal Processing Systems (Special Issue on Multimedia Signal Processing)*, *41*(3), 255-269.

Dybkjaer, L., & Bernsen, N.O . (2001). Usability evaluation in spoken language dialogue system. *Proceedings of the Workshop on Evaluation for Language and Dialogue Systems, Association for Computational Linguistics 39th Annual Meeting and 10th Conference of the European Chapter,*,(pp. 9-18).

Fukumoto, M., Suenaga, Y., & Mase, K. (1994). Finger-pointer: Pointing interface by image processing. *Computer Graphics*, *18(5),* 633-642.

Gong, Y. (1995). Speech recognition in noisy environments: A survey. *Speech Communication*, *16*, 261-291.

Gould, J. & Lewis, C. (1985). Design for usability: Key principles and what designers think. *Communications of the ACM*, *28*(3): 300-301.

Holzman, T. G. (1999). Computer-human interface solutions for emergency medical care. *Interactions, 6(3),* 13-24.

Hone, K. S., & Graham, R. (2001). Subjective assessment of speech system interface usability. *Proceedings of the Eurospeech Conference* (pp. 2083-2086).

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 159-166).

Huang, X., Acero, A., Chelba, C., Deng, L., Droppo, J., Duchene, D., Goodman, J., Hon, H., Jacoby, D., Jiang, L., Loynd, R., Mahajan, M., Mau, P., Meredith, S., Mughal, S., Neto, S., Plumpe, M., Stery, K., Venolia, G., Wang, K., & Wang. Y. (2001). MIPAD: A multimodal interaction prototype. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, *1*, 9-12.

Huang, X., Acero, A., Chelba, C., Deng, L., Duchene, D., Goodman, J., Hon, H., Jacoby, D., Jiang, L., Loynd, R., Mahajan, M., Mau, P., Meredith, S., Mughal, S., Neto, S., Plumpe, M., Wang, K., & Wang, Y. (2000). MIPAD: A next generation PDA prototype. *Proceedings of the International Conference on Spoken Language Processing, 3*, 33-36.

Hsu B.-J., Mahajan M., & Acero A. (2005). *Multimodal text entry on mobile devices.* The ninth bi-annual IEEE workshop on Automatic Speech Recognition and Understanding (Demo). Retrieved July 20, 2006, from

http://research.microsoft.com/~milindm/2005-milindm-ASRU-Demo.pdf

Johnston, M. (1998). Unification-based multimodal parsing. *Proceedings of the International Joint Conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics* (pp. 624-630).

Karat, C.-M., Halverson, C., Horn, D., & Karat, J. (1999). Patterns of entry and correction in large vocabulary continuous speech recognition systems. *Proceedings of the International Conference for Computer-Human Interaction* (pp. 568-575).

Kay, M. (1979). Functional grammar. *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Society* (pp. 142-158).

Kumar, S., & Cohen, P. R. (2000). Towards a fault-tolerant multi-agent system architecture. *Fourth International Conference on Autonomous Agents* (pp. 459-466).

Li, X., Malkin J., & Bilmes, J. (2006). A high-speed, low-resource ASR back-end based on

custom arithmetic. *IEEE Transactions on Audio, Speech and Language Processing*, *14*(5), 1683-1693.

Lockwood, P., & Boudy, J. (1992). Experiments with a nonlinear spectral subtractor (NSS), hidden Markov models and the projection for robust speech recognition in cars. *Speech Communication*, *11*(2-3), 215-28.

Macho, D., Mauuary, L., Noé, B., Cheng, Y. M., Ealey, D., Jouvet, D., Kelleher, H., Pearce, D., & Saadoun, F. (2002). Evaluation of a noise-robust DSR front-end on Aurora databases. *Proceedings of the International Conference on Spoken Language Processing* (pp. 17-20).

Minsky, M. (1975). A framework for representing knowledge. In P. Winston (Ed.), *The psychology of computer vision* (pp 211-277). New York: McGraw-Hill.

Neal, J. G., & Shapiro, S. C. (1991). Intelligent multimedia interface technology. In J. Sullivan & S. Tyler (Eds.), *Intelligent user interfaces* (pp.11-43). New York: ACM Press.

Nielsen, J. (1993). *Usability engineering.* San Diego: Academic Press .

Norman, D. A. & Draper, S. W. (Eds.). (1986). *User-centered system design: New perspectives on human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Oviatt, S. L. (1999). Ten myths of multimodal interaction, *Communications of the ACM, 42* (11), 74-81.

Oviatt, S. L., Bernard, J., & Levow, G. (1999). Linguistic adaptation during error resolution with spoken and multimodal systems. *Language and Speech* (special issue on *Prosody and Conversation), 41*(3-4), 415-438.

Oviatt, S. L., Cohen, P. R., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J., & Ferro, D. (2000). Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. *Human Computer Interaction*, 263-322. (online version), Retrieved July 20, 2006, from http://www.cse.ogi.edu/CHCC/Publications/designing_user_interface_multimodal_speech_oviatt.pdf

Oviatt, S. L., DeAngeli, A., & Kuhn, K. (1997). Integration and synchronization of input modes during multimodal human-computer interaction. *Proceedings of Conference on Human Factors in Computing Systems (CHI'97)* (pp. 415-422).

Oviatt, S. L. & Olsen, E. (1994). Integration themes in multimodal human-computer interaction. In Shirai, Furui, & Kakehi (Eds.), *Proceedings of the International Conference on Spoken Language Processing, 2*, 551-554.

Oviatt, S. L., & vanGent, R. (1996). Error resolution during multimodal human-computer interaction. *Proceedings of the International Conference on Spoken Language Processing, 2*, 204-207.

Pavlovic, V., Berry, G., & Huang, T. S. (1997). Integration of audio/visual information for use in human-computer intelligent interaction. *Proceedings of IEEE International Conference on Image Processing* (pp. 121-124).

Pavlovic, V., & Huang, T. S., (1998). Multimodal prediction and classification on audio-visual features. *AAAI'98 Workshop on Representations for Multi-modal Human-Computer Interaction*, 55-59.

Ravden, S. J., & Johnson, G. I. (1989). *Evaluating usability of human-computer interfaces: A practical method*. Chichester: Ellis Horwood.

Reeves, L. M., Lai, J., Larson, J. A., Oviatt, S., Balaji, T. S., Buisine, S., Collings, P., Cohen, P., Kraal, B., Martin, J. C., McTear, M., Raman, T. V., Stanney, K. M., Su, H., & Wang, Q. Y. (2004). Guidelines for multimodal user interface design. *Communications of the ACM – Special Issue on Multimodal Interfaces*, *47*(1), 57-59.

Rhyne, J. R., & Wolf, C. G. (1993). Recognition-based user interfaces. In H. R. Hartson & D. Hix (Eds.), *Advances in Human-Computer Interaction*, *4,* 191-250.

Roe, D. B., & Wilpon, J. G. (Eds.). (1994). *Voice communication between humans and machines*. Washington, D.C: National Academy Press.

Rubin, P., Vatikiotis-Bateson, E., & Benoit, C. (1998). Special issue on audio-visual speech processing. *Speech Communication*, *26* (1-2).

Schwartz, D. G. (1993). *Cooperating heterogeneous systems: A blackboard-based meta approach*. Unpublished Ph. D. thesis, Case Western Reserve University.

Shaikh, A., Juth, S., Medl, A., Marsic, I., Kulikowski, C., & Flanagan, J. (1997). An architecture for multimodal information fusion. *Proceedings of the Workshop on Perceptual User Interfaces,* 91-93.

Stork, D. G., & Hennecke, M. E. (Eds.) (1995). *Speechreading by humans and machines.* New York: Springer Verlag.

Suhm, B. (1998). *Multimodal interactive error recovery for non-conversational speech user interfaces.* Ph.D. thesis, Fredericiana University.

Tashev, I., & Malvar, H. S. (2005). A new beamformer design algorithm for microphone arrays. *Proceedings of International Conference of Acoustic, Speech and Signal Processing*, *3*, 101-104.

Varga, I., Aalburg, S., Andrassy, B., Astrov, S., Bauer, J.G., Beaugeant, C., Geissler, C., & Hoge, H. (2002). ASR in mobile phones - an industrial approach, *IEEE Transactions on Speech and Audio Processing*, *10*(8), 562- 569.

Vo, M. T., Houghton, R., Yang, J., Bub, U., Meier, U., Waibel, A., & Duchnowski, P. (1995). Multimodal learning interfaces. *Proceedings of the DARPA Spoken Language Technology Workshop.* Retrieved July 20, 2006 from

http://www.cs.cmu.edu/afs/cs.cmu.edu/user/tue/www/papers/slt95/paper.html

Vo, M. T., & Wood, C. (1996). Building an application framework for speech and pen input integration in multimodal learning interfaces. *Proceedings of IEEE International Conference of Acoustic, Speech and Signal Processing, 6*, 3545-3548.

Wang, J. (1995). Integration of eye-gaze, voice and manual response in multimodal user interfaces. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* (pp. 3938-3942).

Wang, K. (2004). A detection based approach to robust speech understanding. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1,* 413-416.

Wang, Y. (2001). Robust language understanding in MiPAD. *Proceedings of the Eurospeech Conference* (pp. 1555-1558).

Wang, Y., Acero, A., & Chelba, C. (2003). Is word error rate a good indicator for spoken language understanding accuracy? *Proceedings of the Workshop on Automatic Speech Recognition Workshop and Understanding* (pp 577-582).

Wu, L., Oviatt, S., &. Cohen, P. (1999). Multimodal integration-A statistical view. *IEEE Transactions on Multimedia*, *1*(4), 334-341.

Yu, D., Ju, Y. C., Wang, Y., & Acero, A. (2006). N-gram based filler model for robust grammar authoring. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing1,* 565-569.

Yu, D., Wang, K., Mahajan, M., Mau, P., & Acero, A. (2003). Improved name recognition with user modeling. *Proceedings of Eurospeech* (pp. 1229-1232).

KEY TERMS

Modality: A communication channel between human and computer, such as vision, speech, keyboard, pen, and touch.

Modality Fusion: A process of combining information from different input modalities in a principled way. Typical fusion approaches include early fusion, in which signals are integrated at the feature level, and late fusion, in which information is integrated at the semantic level.

Multimodal User Interface: A user interface with which users can choose to interact with a system through one of the supported modalities, or multiple modalities simultaneously, based on the usage environment or preference. Multimodal user interface can increase the usability because the strength of one modality often compensates for the weaknesses of another.

Push to Talk: A method of modality switching where a momentary button is used to activate and deactivate the speech recognition engine.

Speech-Centric Multimodal User Interface: A multimodal user interface where speech is the central and primary interaction modality.

Typed feature Structure: An extended, recursive version of attribute-value type data structures, where a value can, in turn, be a feature structure. It indicates the kind of entity it represents with a type, and the values with an associated collection of feature-value or attribute-value pairs. In the typed feature structure, a value may be nil, a variable, an atom, or another typed feature structure.

User-Centered Design: A design philosophy and process in which great attention is given to the needs, expectations, and limitations of the end user of a human-computer interface at each stage of the design process. In the user-centered design process, designers not only analyze and foresee how users are likely to use an interface, but also test their assumptions with actual users under real usage scenario.