Designing the User Interface for Multimodal Speech and Pen-based Gesture Applications: State-of-the-Art Systems and Future Research Directions

Sharon Oviatt, *Oregon Graduate Institute of Science and Technology,*
Phil Cohen, *Oregon Graduate Institute,* **Lizhong Wu,** *HNC Software,* **John Vergo,** *IBM T. J. Watson Research,* **Lisbeth Duncan,** *The Boeing Company,*
Bernhard Suhm, *BBN Technologies,* **Josh Bers,** *BBN Technologies,*
Thomas Holzman, *NCR,* **Terry Winograd,** *Stanford University,*
James Landay, *University of California at Berkeley,*
Jim Larson, *Intel,* **& David Ferro,** *Unisys*

ABSTRACT

The growing interest in multimodal interface design is inspired in large part by the goals of supporting more transparent, flexible, efficient, and powerfully expressive means of human-computer interaction than in the past. Multimodal interfaces are expected to support a wider range of diverse applications, to be usable by a broader spectrum of the average population, and to function more reliably under realistic and challenging usage conditions. In this paper, we summarize the emerging architectural approaches for interpreting speech and pen-based gestural input in a robust manner— including early and late fusion approaches, and the new hybrid symbolic/statistical approach. We also describe a diverse collection of state-of-the-art multimodal systems that process users' spoken and gestural input. These applications range from map-based and virtual reality systems for engaging in simulations and training, to field medic systems for mobile use in noisy environments, to web-based transactions and standard text-editing applications that will reshape daily computing and have a significant commercial impact. To realize successful multimodal systems of the future, many key research challenges remain to be addressed. Among these challenges are the development of cognitive theories to guide multimodal system design, and the development of effective natural language processing, dialogue processing, and error handling techniques. In addition, new multimodal systems will be needed that can function more robustly and adaptively, and with support for collaborative multi-person use. Before this new class of systems can proliferate, toolkits also will be needed to promote software development for both simulated and functioning systems.

CONTENTS

1. INTRODUCTION TO MULTIMODAL SPEECH AND GESTURE INTERFACES

The growing interest in multimodal interface design is inspired largely by the goal of supporting more transparent, flexible, efficient, and powerfully expressive means of human-computer interaction. Multimodal interfaces also are expected to be easier to learn and use, and are preferred by users for many applications. They have the potential to expand computing to more challenging applications, to be used by a broader spectrum of everyday people, and to accommodate more adverse usage conditions than in the past. This class of systems represents a relatively new direction for computing that draws from the myriad input and output technologies currently becoming available.

Sharon Oviatt is a professor of Computer Science in the Center for Human-Computer Communication (CHCC) at the Oregon Graduate Institute of Science and Technology (OGI); her current interests include human-computer interfaces for multimodal systems, spoken language systems, and mobile technology. **Phil Cohen** is a professor of Computer Science in CHCC at OGI with interests in multimodal interaction, multi-agent systems, computational linguistics, and artificial intelligence. **Lizhong Wu** is an information engineer with an interest in multimedia signal processing and multimodal recognition; he is a senior staff scientist focused on machine learning in the advanced technology solutions group at HNC Software. **John Vergo** is a computer scientist with an interest in HCI, especially speech, natural language understanding and multimodal interaction; he is a research staff manager in the applications and advanced human computer interaction group of IBM T.J. Watson Research. **Lisbeth Duncan** is a computer scientist with an interest in natural language understanding and human-computer interfaces; she is an associate technical fellow in the natural language processing group in Boeing's Mathematics and Computing Technology Division. **Bernhard Suhm** is a computer scientist with an interest in speech recognition and multimodal human-computer interfaces; he is a scientist in the speech and language processing group of BBN Technologies. **Josh Bers** is a computer scientist with an interest in multimodal user interfaces; he is an engineering manager in the speech solutions group of GTE's Technology Organization. **Thomas Holzman** is a cognitive psychologist who applies user-centered systems engineering to the design and evaluation of multimedia computer-human interfaces; he is the director of cognitive engineering in the Corporate Technology organization at NCR Corporation. **Terry Winograd** is a professor of computer science at Stanford University with interests in HCI, collaborative computing, and ubiquitous computing. **James Landay** is an assistant professor of computer science at the University of California at Berkeley, with current interests in HCI, pen-based sketching, and software tools for developing multimodal systems. **James Larson** is manager of advanced human I/O at the Intel Architecture Labs with an interest in speech and multimedia applications. **David Ferro** is a computer scientist in the natural language understanding group of Unisys, with interests in speech and contextualized HCI.

Since Bolt's (1980) original "Put That There" concept demonstration, which processed multimodal speech and manual pointing during object manipulation, considerable strides have been made in developing more general multimodal systems that process complex gestural input other than just pointing, examples of which will be outlined in section 4 of this paper. Since Bolt's early inspiration, the basic architectural components and framework needed to support more general multimodal systems have become better established, as will be described in section 3. In contrast to Bolt's initial concept, which was a limited prototype, significant progress also has occurred in building a variety of real applications, five of which are illustrated in section 4. In addition, during the past decade proactive empirical work has contributed predictive information on human-computer multimodal interaction, which has provided a foundation for guiding the design of new multimodal systems that are still in the planning stages.

In a more general vein, major advances in new input technologies and algorithms, hardware speed, distributed computing, and spoken language technology in particular, all have supported the emergence of more transparent and natural communication with this new class of multimodal systems. During the past decade, due largely to progress inspired by the DARPA Speech Grand Challenge project and similar international efforts (Martin et al., 1997; Cole et al., 1997), there has been significant progress in the development of spoken language technology (SLT). Spoken language systems now are implemented extensively for telephony applications (Spiegel & Kamm, 1997), and on workstations, and they are beginning to appear on small palm computers. These new technical capabilities, along with advances in natural language processing, are leading to increasingly conversational query-and-answer systems. Spoken language systems also are supporting new training systems for learning foreign languages and basic reading skills, as well as the commercialization of automated dictation systems for applications such as medical charting, legal records, and word processing.

Like spoken language technology, steady advances have occurred in pen-based hardware and software capabilities, which currently provide handwriting and gesture recognition on handhelds, small pocket-sized computers, and now are beginning to appear on mobile phones. Typically, these pen-based applications are used to automate telephony, or to extend personal memory during management of calendars, contact information, and other personal information. Pen computing also supports visual-spatial applications involving map-based interaction, as well as specialized sketching applications for the design of flow charts, user interface designs, circuit designs, and the like. These strides in pen technology, spoken language systems, and the development of increasingly general and robust multimodal systems all are clear landmarks of progress since Put That There's initial demonstration of combined speech and manual gesturing in the user interface.

In this paper, we begin in section 2 by introducing multimodal speech and pen-based gesture interfaces, with a focus on their primary advantages and optimal uses. To date, multimodal systems that combine either speech and pen input (Oviatt & Cohen, 2000) or speech and lip movements (Benoit, et al, in press; Stork & Hennecke, 1995; Rubin, Vatikiotis-Bateson, & Benoit, 1998) constitute the two major research areas within the field. Although many of the issues discussed for multimodal systems incorporating speech and 2-D pen gestures also are relevant to those involving continuous 3-D manual gesturing (e.g., Bolt's system), the latter type of system presently is less mature. This primarily is because of the significant problems associated with segmenting and interpreting continuous manual movements, compared with a stream of x,y ink coordinates. As a result of this difference, the multimodal subfield involving speech and pen-based gestures has been able to explore a wider range of research issues and to advance more rapidly in its multimodal architectures.

In section 3, we summarize the architectural approaches currently being used to interpret dual input signals — including early and late fusion approaches, and a new hybrid symbolic/statistical approach to speech and pen-based gesture interpretation. In section 4, we then illustrate five diverse state-of-the-art multimodal systems that support challenging applications. These include map-based and virtual reality systems for engaging in simulations and training (sections 4.1 and 4.3), text-editing and web-based catalogue ordering that have the potential to reshape daily computing for everyday users (sections 4.2 and 4.5), and mobile field-medic systems for documenting trauma care while ambulance crews work in noisy and chaotic multi-person settings (section 4.4). Finally, in section 5 we discuss the many multimodal research challenges that remain to be addressed.

2. ADVANTAGES AND OPTIMAL USES OF MULTIMODAL INTERFACE DESIGN

As applications generally have become more complex, a single modality does not permit the user to interact effectively across all tasks and environments (Larson, Ferro, & Oviatt, 1999). A multimodal interface offers the user freedom to use a combination of modalities, or to switch to a better-suited modality, depending on the specifics of the task or environment. Since individual input modalities are well suited in some situations, and less ideal or even inappropriate in others, modality choice is an important design issue in a multimodal system. In this section, we discuss the strengths of speech and pen input as individual modalities, as well as issues specific to their benefits within a combined multimodal interface.

Among other things, speech input offers speed, high-bandwidth information, and relative ease of use. It also permits the user’s hands and eyes to be busy with a task, which is particularly valuable when users are in motion or in natural field settings. Users tend to prefer speech for functions like describing objects and events, sets and subsets of objects, out-of-view objects, conjoined information, past and future temporal states, as well as for issuing commands for actions or iterative actions (Cohen & Oviatt, 1995; Oviatt & Cohen, 1991). During multimodal pen/voice interaction, users tend to prefer entering descriptive information via speech, although their preference for pen input increases for digits, symbols, and graphic content (Oviatt & Olsen, 1994; Oviatt, 1997; Suhm, 1998).

While also supporting portability, pen input has a somewhat different and multi-functional range of capabilities. Although the pen can be used to write words that are analogous to speech, it also can be used to convey symbols and signs (e.g., digits, abbreviations), gestures, simple graphics and artwork, and to render signatures. In addition, it can be used to point, to select visible objects like the mouse does in a direct manipulation interface, and as a means of microphone engagement for speech input. From a usage standpoint, pen input provides a more private and socially-acceptable form of input in public settings, and a viable alternative to speech under circumstances of extreme noise (Holzman, 1999; Gong, 1995). In architectural and similar domains, sketching and drawn graphics are a particularly rich and generative capability (Landay, 1996). In addition, pen input to maps and other graphic displays can easily and efficiently convey spatial information about precise points, lines, and areas (Oviatt, 1997). In brief, the pen offers critical capabilities for interacting with any form of graphic application, and it potentially can provide a very versatile and opportune base system, especially for mobile tasks.

As forms of human language technology, spoken and pen-based input have the advantage of permitting users to engage in more powerfully expressive and transparent information-seeking dialogues. Together, the speech and pen modes can easily be used to provide flexible descriptions of objects, events, spatial layouts, and their interrelation. This is largely because spoken and pen-based input provide complementary capabilities. For example, analysis of the linguistic content of users' integrated pen/voice constructions has revealed that basic subject, verb, and object constituents almost always are spoken, whereas those describing locative information invariably are written or gestured (Oviatt, DeAngeli, & Kuhn, 1997). This complementarity of spoken and gestural input also has been identified as a theme during interpersonal communication (McNeill, 1992).

Compared with speech-only interaction, empirical work with users during visual-spatial tasks has demonstrated that multimodal pen/voice interaction can result in 10% faster task completion time, 36% fewer task-critical content errors, 50% fewer spontaneous disfluencies, and also shorter and more simplified linguistic constructions with fewer locative descriptions (Oviatt, 1997; Oviatt & Kuhn, 1998). This constellation of multimodal performance advantages corresponds with a 90-100% user preference to interact multimodally (Oviatt, 1997; Oviatt, Cohen, & Wang, 1994). In large part, people's performance difficulties during visual-spatial tasks are due to their error-proneness and reluctance to articulate spatially oriented information. During multimodal interaction, people instead prefer to use pen input to point or create graphics, since it generally is a more effective and precise way to convey locations, lines, areas, and other inherently spatial information. Likewise, when people are manipulating 3D objects, a multimodal interface that permits them to speak and gesture while handling objects manually is both preferred and more efficient (Hauptmann, 1989).

A particularly advantageous feature of multimodal interface design is its ability to support superior error handling, compared with unimodal recognition-based interfaces, both in terms of error avoidance and graceful recovery from errors (Oviatt & vanGent, 1996; Oviatt, Bernard, & Levow, 1999; Oviatt, 1999a; Rudnicky & Hauptmann, 1992; Suhm, 1998). There are both user-centered and system-centered reasons why multimodal systems facilitate error recovery. First, in a multimodal interface users will select the input mode that they judge to be less error prone for particular lexical content, which leads to error avoidance (Oviatt & vanGent, 1996). Second, users' language is simplified when interacting multimodally, which reduces the complexity of natural language processing and thereby reduces recognition errors (Oviatt & Kuhn, 1998). Third, users have a strong tendency to switch modes after system errors, which facilitates error recovery (Oviatt, Bernard, & Levow, 1999). The fourth reason why multimodal systems support more graceful error handling is that users report less subjective frustration with errors when interacting multimodally, even when errors are as frequent as in a unimodal interface (Oviatt & vanGent, 1996). Finally, a well-designed multimodal architecture can support *mutual disambiguation* of input signals. Mutual disambiguation involves recovery from unimodal recognition errors within a multimodal architecture, because semantic information from each input mode supplies partial disambiguation of the other mode— thereby leading to more stable and robust overall system performance (Oviatt, 1999a).[1] To reap these error-handling advantages fully, a multimodal system must be designed so that the speech and pen modes provide parallel or duplicate functionality, which means that users can accomplish their goals using either mode.

Since there are large individual differences in ability and preference to use different modes of communication, a multimodal interface permits the user to exercise selection and control over how they interact with the computer (Fell et al., 1994; Karshmer & Blattner, 1998). In this respect, multimodal interfaces have the potential to accommodate a broader range of users than traditional graphical user interfaces (GUIs) and unimodal interfaces— including users of different ages, skill levels, native language status, cognitive styles, sensory impairments, and other temporary or permanent handicaps or illnesses. For example, a visually impaired user may prefer speech input

[1] The phenomenon of mutual disambiguation is analogous to what has been called "super-additivity effects" in the multimodal speech and lip literature (Iverson, Bernstein, & Auer, 1998), which also are associated with improved recognition rates.

and text-to-speech output, as may a manually impaired user (e.g., with repetitive stress injury, or their arm in a cast). In contrast, a user with a hearing impairment, strong accent, or a cold may prefer pen input. Likewise, a young preschooler using an educational application can use *either* speech or graphical pen input well before a keyboard is a viable input device. A multimodal interface also permits alternation of individual input modes, which can be critical in preventing overuse and physical damage to any single modality, especially during extended periods of computer use (Markinson, 1993).

Interfaces involving spoken or pen-based input, as well as the combination of both, are particularly effective for supporting mobile tasks, such as communications and personal navigation. Unlike the keyboard and mouse, both speech and pen are compact and portable. When combined, people can shift these input modes from moment to moment as environmental conditions change (Holzman, 1999). There is a sense in which mobility can induce a state of "temporary disability," such that a person is unable to use a particular input mode for some period of time. For example, a user carrying a child may be temporarily unable to use pen or touch input at a public information kiosk, although speech is unaffected. In this respect, a multimodal pen/voice system permits the alternation needed to cover a wider range of changing environmental circumstances that may be encountered during actual field use.

In summary, it has been demonstrated that combined multimodal speech and gesture interfaces:

- Permit flexible use of input modes, including alternation and integrated use
- Support improved efficiency, especially when manipulating graphical information
- Can support less disfluent, shorter, and more linguistically-simplified constructions than a speech-only interface, which results in more robust natural language processing

- Support greater precision of spatial information than a speech-only interface, because pen input conveys rich and precise graphical information
- Satisfy higher levels of user preference
- Support enhanced error avoidance and ease of error resolution
- Accommodate a wider range of users, tasks, and environmental situations
- Are adaptable during the continuously changing environmental conditions of mobile use
- Accommodate individual differences, such as permanent or temporary handicaps
- Prevent overuse of any individual mode during extended computer usage

3. ARCHITECTURAL APPROACHES TO MULTIMODAL INTEGRATION AND SYSTEMS

As an introduction to the multimodal system descriptions that follow in section 4, in this section we summarize the main architectural requirements and components of multimodal systems. In particular, the main architectural approaches are outlined for interpreting multimodal speech and pen-based gestures in a robust manner—including primarily late semantic fusion approaches, but also the introduction of a new hybrid symbolic/statistical approach that illustrates the future direction of multimodal architectures.

3.1. Introduction to Multimodal Architectural Requirements

Many early multimodal interfaces that handled combined speech and gesture, such as Bolt’s Put That There system (Bolt, 1980), were based on a control structure in which multimodal integration occurred during the process of parsing spoken language. When the user spoke a deictic expression, such as “here” or “this”, the system would search for a synchronized gestural act that designated the spoken referent. While such an approach suffices for processing a *point-and-speak multimodal integration pattern*, unfortunately less than 20% of all users multimodal commands are of this limited type (Oviatt, DeAngeli, & Kuhn, 1997; McNeill, 1992). For this reason, multimodal pen/voice systems must be able to process richer pen-based input than just pointing — including gestures (e.g., arrows, delete marks), digits, symbols, simple graphic marks (e.g., square to designate a building), and so forth. To support the development of broadly functional multimodal systems, a more general processing architecture clearly is needed. Ideally, such an architecture should handle both (1) a variety of multimodal speech-and-gesture integration patterns, and also (2) the interpretation of unimodal gestural or spoken input, as well as combined multimodal input. Such an architecture would support the development of multimodal systems with multiple modalities that are used and processed individually as *input alternatives* to one another, as well as those designed to support *combined multimodal input* from two or more modes.

For multimodal systems designed to handle joint processing of input signals, there are two main subtypes of multimodal architectures — ones that integrate signals at the *feature* level (i.e., “early fusion”) and others that integrate information at a *semantic* level (i.e., “late fusion”). Systems that utilize the early feature-fusion approach generally are based on multiple Hidden Markov Models or temporal neural networks[2]. Examples of representative research include Bregler et al. (1993), Vo et al. (1995), Pavlovic, Berry & Huang (1997), and Pavlovic & Huang (1998). In a feature-fusion architecture, the recognition process in one mode influences the course of recognition in the other. Feature fusion generally is considered more appropriate for closely coupled and synchronized modalities, such as speech and lip movements (Stork & Hennecke, 1995; Rubin, Vatikiotis-Bateson, & Benoit, 1998), for which both input channels provide corresponding

[2] Hidden Markov Modeling (HMM) is a state-of-the-art statistical modeling technique that has been widely applied to problems such as large-vocabulary continuous speech recognition and handwriting recognition (Rabiner, 1989). Neural networks (NNs) are an alternative pattern recognition technique, and temporal neural networks (TNNs) are ones capable of modeling the temporal structure of input signals, such as time-delay neural networks (Waibel et al., 1989) and recurrent neural networks (Elman, 1990).

information about the same articulated phonemes and words. However, such a system tends not to apply or generalize as well if it consists of modes that differ substantially in the information content or time scale characteristics of their features. This is the case, for example, with speech and pen input, for which the input modes provide different but complementary information that is typically integrated at the utterance level. Modeling complexity, computational intensity, and training difficulty are typical problems associated with the feature-fusion integration approach. For example, a large amount of training data is required to build this type of system. Unfortunately, multimodal training corpora rarely have been collected, and currently are at a premium.

Generally, multimodal systems using the late semantic fusion approach include individual recognizers and a sequential integration process. These individual recognizers can be trained using unimodal data, which are easier to collect and already publicly available for modalities like speech and handwriting. This type of architecture also can leverage from existing and relatively mature unimodal recognition techniques and off-the-shelf recognizers. Such unimodal systems often can be integrated directly, or changed when necessary without re-training. In this respect, systems based on semantic fusion can be scaled up easier, whether in the number of input modes or the size and type of vocabulary set. Examples of representative studies and systems that have used semantic fusion include Put That There (Bolt, 1980), ShopTalk (Cohen, et al., 1989), QuickSet (Cohen et al., 1997), CUBRICON (Neal & Shapiro, 1991), Virtual World (Codella et al., 1992), Finger-Pointer (Fukumoto, Suenga, & Mase, 1994), VisualMan (Wang, 1995), and Jeanie (Vo & Wood, 1996).

Multimodal systems that are designed to process combined input from two or more modes also require an architecture that supports fine-grained time stamping of at least the beginning and end of each input signal. Since users' multimodal speech and gesture constructions can involve either *sequentially-integrated* or *simultaneously delivered* signal pieces, a multimodal architecture also must be prepared to handle input signals that may or may not be overlapped in their temporal delivery (Oviatt, DeAngeli, & Kuhn, 1997). Empirical work on speech and gesture input has established that users' written input precedes speech during a sequentially-integrated multimodal command (Oviatt, DeAngeli, & Kuhn, 1997), and it also has clarified the distribution of typical inter-modal lags. This type of information has been useful in determining whether two signal pieces are part of a multimodal construction, or whether they should be interpreted as unimodal commands. In addition, data on inter-modal lags has been used to establish the temporal thresholds for joining signal pieces in multimodal architectures (see section 4.1).

One major design goal of multimodal systems is the selection of complementary input modes that are capable of forging a highly synergistic overall combination. In theory, a well designed multimodal system should be able to integrate complementary modalities such that the strengths of each modality are capitalized upon and used to overcome weaknesses in the other (Cohen et al., 1989; Oviatt et al., 1992). This general approach can result in a more broadly functional system, as well as a more reliable one, in part due to *mutual disambiguation.* In fact, empirical work has demonstrated that a well-integrated multimodal system can yield significant levels of mutual disambiguation between input signals (i.e., with speech disambiguating the meaning of gesture, and vice versa). Mutual disambiguation generates higher overall recognition rates and more stable system functioning than either component technology can as a stand-alone (Oviatt, 1999a, 2000).

In summary, to create useful and general multimodal pen/voice architectures that are capable of processing both separate and combined input modes in a robust manner ideally requires:

- Parallel recognizers and interpreters that produce a set of time-stamped meaning fragments for each continuous input stream
- A common framework for representing meaning fragments derived from multiple modalities
- A time-sensitive grouping process that determines when to combine individual meaning fragments from each modality stream
- Meaning fusion operations that combine semantically- and temporally-compatible meaning fragments
- A data-driven statistical process that enhances the likelihood of selecting the best joint interpretation of multimodal input
- A flexible asynchronous architecture that permits multiprocessing, keeps pace with user input, and potentially handles input from multiple simultaneous users
- A multimodal interface design that combines complementary modes in a synergistic manner, thereby yielding significant levels of mutual disambiguation between modes and improved recognition rates

3.2. Multi-agent Architectures and Multimodal Processing Flow

Before discussing the motivation and specifics of multimodal speech and pen-based gesture architectures, it is important to identify the primary ways in which emerging multimodal architectures are distinct from those of standard GUIs. First, GUIs typically assume that there is a single event stream that controls the underlying event loop. For example, most GUIs will ignore typed input while a mouse button is depressed. However, for many multimodal interfaces the need to process *simultaneous input* from different streams will be the norm. Second, GUIs assume that the basic interface actions, such as selection of an item, are atomic and unambiguous events. In contrast, multimodal systems are being designed to process natural human input modes via recognition-based technologies, which must handle uncertainty and therefore are based on *probabilistic* methods of processing. Third, GUIs often are built to be separable from the application software that they control, but the interface components themselves usually reside together on the same machine. In contrast, recognition-based user interfaces typically have larger computational and memory requirements, which can make it preferable to *distribute the interface* over a network such that separate machines handle different recognizers. For example, cell phones and networked PDAs may extract features from speech input and transmit them to a recognizer that resides on a server, as does BBN's Portable Voice Assistant (see section 4.5). In light of these new architectural requirements, multimodal research groups currently are rethinking and redesigning basic user interface architectures.

Figure 1 depicts the basic information flow needed to process multimodal speech and gesture input.[3] In such an architecture, speech and pen-based gestures are recognized in parallel, and each is processed by an understanding component. The results are meaning representations that are fused by the multimodal integration component, which also is influenced by the system's dialogue management and interpretation of current context. During the integration process, alternative lexical candidates for the final multimodal interpretation are ranked according to their probability estimates on an n-best list. The best ranked multimodal interpretation then is sent to the application invocation and control component, which transforms this information into a series of

[3] Since the present paper concentrates on multimodal input, we have not displayed a more complete information processing architecture for a full multimodal dialogue system. Instead, we include here only schematic dialogue management and response planning components.

Note also that the most common variant of this information processing flow for multimodal pen/voice systems is cases in which gesture functionality is limited to deictic pointing. For such systems, speech dominates the natural language processing (NLP) and deictic points are filled into the speech frame before NLP occurs. No separate language processing is performed on the pen-based pointing gestures.

commands to one or more “back end” application system(s). System output typically is presented via either a graphical, auditory (e.g., for telephony), or multimedia display. Both system context and dialogue management typically are altered during user input, as well as during system output generation.

[INSERT FIGURE 1 HERE]

There are numerous ways to realize this information processing flow as an architecture. One well-understood way is to pipeline the components via procedure calls or, if the system is distributed but homogeneous in its programming language, remote procedure calls. However, if the system is heterogeneous (e.g., in programming languages, operating systems, or machine characteristics), the above method may prove difficult. To provide a higher level layer that supports distributed heterogeneous software, while shielding the designer from the details of communication, a number of research groups have developed and employed *multi-agent architectures*, such as the Open Agent Architecture[4] (Cohen, et al., 1994; Martin, Cheyer, & Moran, 1999) and Adaptive Agent Architecture (Kumar & Cohen, 2000).

In a multi-agent architecture, components may be written in different programming languages, on different machines, and with different operating systems. Each component is “wrapped” by a layer of software that enables it to communicate via a standard language over TCP/IP. The resulting component-with-communication-layer is termed an *agent*. The agent communication language often uses message types reminiscent of speech act theory, but ones extended to handle asynchronous delivery, triggered responses, multi-casting, and other concepts from distributed systems. In some multi-agent architectures, agents communicate directly with other components about which they have knowledge (e.g., component's name, API). This design has the advantage of no intermediaries, but it can be brittle in the face of agent failure.

As an alternative design, many architectures have adopted a *facilitated* form of communication, in which agents do not need to know to whom they are making requests or supplying answers. Instead, these agents communicate through a known facilitator, which routes messages to the interested and capable receivers. This becomes a *hub-and-spoke architecture,* with all agents communicating via the central facilitator. The facilitator provides a place for new agents to connect at run time, and they then can be discovered by other agents and incorporated into the ongoing distributed computation. The hub also becomes a locus for building collaboration systems, since the facilitator can route communications to multiple agents that may be interested in the same messages.

Figure 2 illustrates the same basic components as Figure 1, but now arrayed around a central facilitator. When that facilitator also has a global area in which to store data, it is sometimes referred to as a *blackboard system* (Erman & Lesser, 1975; Schwartz, 1993). Note that a facilitator/hub can be a bottleneck, possibly impeding high-volume multimedia data transfer (e.g., speech). It also is a single point of failure, which can lead to a lack of system robustness. Recent research has developed fault-tolerant multiagent architectures (Kumar & Cohen, 2000), which employ a team of facilitators that can share the load in case of individual facilitator failure.

[INSERT FIGURE 2 HERE]

[4] The Open Agent Architecture (OAA) is a trademark of SRI International.

Within this type of facilitated architecture, speech and gesture input can arrive in parallel or asynchronously via individual modality agents, and they then are recognized and the results are passed to the facilitator. These results, typically an n-best list of conjectured lexical items as well as time stamp information, then are routed to agents that have advertised the capability of handling this type of data. Next, sets of meaning fragments derived from the speech and pen signals arrive at the multimodal integrator. This agent decides whether and how long to wait for recognition results from other modalities, based on the system's temporal thresholds. It then attempts to fuse the meaning fragments into a semantically- and temporally-compatible whole interpretation before passing the results back to the facilitator. At this point, the system's final multimodal interpretation is confirmed by the user interface, and executed by any relevant applications. Meanwhile, new input that may have arrived from the same user or other users is processed asynchronously. Processing within this kind of a multi-agent framework is usually bottom-up and asynchronous, with an emergent control flow.[5]

3.3. Frame-based and Unification-based Multimodal Integration

The core of multimodal integration systems based on semantic fusion comprises algorithms that integrate meaning representations derived from speech, gesture and other modalities into a combined overall interpretation. As mentioned earlier, the semantic fusion operation requires that there be a meaning representation framework that is common among modalities, and a well-defined operation for combining partial meanings. During the initial development of multimodal speech and gesture systems, each system had its own meaning representation framework, and an

[5] The current DARPA Communicator project (Goldschen & Loehr, 1999), which is based on the MIT Galaxy Architecture (Seneff, et al., 1998), is an example of a hub-and-spoke architecture. However, its information flow is *scripted* rather than emergent. That is, the hub requires a script that dictates what to do with information of various types as it is received by the system. This means that agents on the network cannot be incorporated without the scriptwriter's knowledge of their existence, such that information flow is not automatically reconfigurable during processing. In the Communicators architecture, the hub also is a single point of failure.

idiosyncratic algorithm for multimodal integration. However, in recent years a data structure called *frames* (Minsky, 1975) or *feature structures* (Kay, 1979) has de facto become accepted to represent meaning. These structures represent objects and relations as consisting of nested sets of attribute/value pairs. Feature structures go beyond frames in their use of shared variables to indicate common substructures.

In order to fuse information derived from multiple modalities, various research groups (Vo & Wood, 1996; Cheyer & Julia, 1995; Pavlovic & Huang, 1998; Shaikh et al., 1997) have independently converged on a strategy of recursively matching and merging attribute/value structures, although details of the algorithms differ. However, the most well developed literature on this topic comes from computational linguistics, in which formal logics of typed feature structures have been developed (Carpenter, 1990, 1992; Calder, 1987). These structures are pervasive in natural language processing, and have been used for lexical entries, grammar rules, and meaning representations (Sag & Wasow, 1999). For feature structure logics, the primary operation is *unification*— a more general approach that subsumes other frame-merging strategies. Unification-based architectures have only recently been applied to multimodal system design (Cohen et al., 1997; Johnston et al., 1997; Wu et al., 1999).

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. Feature-structure unification is a generalization of term unification in logic programming languages, such as Prolog. A feature structure (FS) consists of a type, which indicates the kind of entity it represents, and an associated collection of feature-value or attribute-value pairs, such that a value may be nil, a variable, an atom, or another feature structure. When two feature structures, FS1 and FS2, are unified with respect to a type hierarchy, the types must be in transitive closure of the subtype relation. The values of identical attributes or features also are matched such that, if they are atoms, they must be identical. If one is a variable, it becomes bound to the value of the corresponding feature in the other feature structure. If both are variables, they become bound together, which constrains them to always receiving the same value.

Feature structures are partial in that if no attributes from FS1 correspond to a given attribute (ATTR) in FS2, then the resulting unified FS simply will contain ATTR and its value. If the values are themselves feature structures, the unification operation is applied recursively.[6] Importantly, feature-structure unification can result in a directed acyclic graph structure when more than one value in the collection of feature/values pairs makes use of the same variable. Whatever value is ultimately unified with that variable therefore will fill the value slots of all the corresponding features. Most of the frame-merging integration techniques do not include this capability.

Feature-structure unification is ideally suited to the task of multimodal speech and gesture integration, because unification can combine complementary input from both modes or redundant input, but it rules out contradictory input (Johnston et al., 1997). The basic unification operations can be augmented with constraint-based reasoning to operate declaratively in a multimodal integrator (Johnston, 1998). Given this foundation for multimodal speech and gesture integration, more research still is needed on statistical ranking and filtering of the feature structures to be unified, and on the development of canonical meaning representations that are common across input modes and research sites.

[6] Feature-structure unification differs from term unification in logic programming in that the features are positionally encoded in a term, whereas they are explicitly labeled and unordered in a feature structure. Otherwise, the unification of values is identical.

3.4. New Hybrid Architectures: An Illustration

When statistical processing techniques are combined with a symbolic unification-based approach that merges feature structures, then the multimodal architecture that results is a *hybrid symbolic/statistical one*. Hybrid architectures represent one major new direction for multimodal system development[7]. Such architectures are capable of achieving very robust functioning, compared with a late-fusion symbolic approach alone. In this section we illustrate the general nature and advantages of a hybrid approach by describing the first such architecture developed for a pen/voice multimodal system — QuickSet — which utilizes the Associative Map and Members-Teams-Committee techniques (Wu, Oviatt, & Cohen, 1999; Wu, Oviatt, & Cohen, in submission). QuickSet will be discussed more fully in section 4.1.

For a multimodal speech and gesture system with a semantic fusion architecture, the primary factors that influence recognition performance include: (1) recognition accuracy of the individual modes, (2) the mapping structure between multimodal commands and their speech/pen constituents, (3) the manner of combining posterior probabilities, and (4) the prior distribution of multimodal commands. The Associative Mapping and Members-Teams-Committee (MTC) techniques provide a statistical approach to developing and optimizing the second and third factors, respectively.

For a given application, the *Associative Map* is an architectural component that defines all semantically meaningful mapping relations that exist between the set of speech constituents and the set of pen-based gesture constituents for each multimodal command, since a constituent in one

[7] Multimodal architectures also can be hybrids in the sense of combining Hidden Markov Models (HMMs) and Neural Networks (NNs). This can be an opportune combination in the case of a pen/voice system, since speech is processed well with HMMs and NNs handle pen input well.

mode typically associates with only a limited number of constituents in the other mode. During multimodal recognition, the defined Associative Map supports a simple process of table lookup. This table can be defined directly by a user, or it can be built automatically using labeled data. The Associative Map basically functions to rule out consideration of those speech and gesture feature structures that cannot possibly be unified semantically. As such, it provides an efficient means of quickly ruling out impossible unifications.

Members-Teams-Committee (MTC) is a novel hierarchical recognition technique, the purpose of which is to weight the contributions derived from speech and gesture recognizers based on their empirically-derived relative reliabilities, and in a manner that optimizes system robustness. As illustrated in Figure 3, the MTC is comprised of a three-tiered divide-and-conquer architecture with multiple members, multiple teams, and a committee. The *members* are the individual recognizers that provide a diverse spectrum of recognition results in the form of local posterior estimates[8]. 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.

MTC serves two purposes within the QuickSet system. It is used as a gesture recognizer, and in this case the members are recognizers for the 190 possible gestures. It also is used in QuickSet as

[8] A local posterior estimate is a conditional posterior probability that estimates the conditional probability of a specific recognition result, given the input.

the statistical mechanism for combining recognition scores from the unifiable speech and gesture feature structures.

[INSERT FIGURE 3 HERE]

The Associative Map and MTC techniques provide an approach to refining the multimodal integration process so that different weights are assigned to different modes and different constituents, thereby enhancing overall system robustness. The primary difference between this integration approach and the conventional approach is that in conventional approaches the probability of the merged feature structures is the cross-product of the probabilities of individual feature structures. In the approach presented here, the probability of the merged feature structure is the weighting interpolation of the probabilities of individual feature structures. The weighting parameters and their confidence levels then are estimated using the MTC technique.

The multimodal hybrid architecture summarized in this section recently has been evaluated using a multimodal corpus and the QuickSet system (Oviatt, 1999a), and has achieved 95.26% correct recognition performance — or within 1.4% of the theoretical system upper bound (Wu, Oviatt, & Cohen, 1999)[9]. The favorable performance of the MTC architectural approach can be attributed to a variety of factors, including the discriminative training scheme[10] for learning weighting parameters (i.e., which maximizes the correct recognition hypothesis), training on multiple sets of weighting parameters (i.e., for smoother estimates and improved generalizability of performance),

[9] From a realistic engineering perspective, individual recognition-based systems will never perform at a 100% correct level (e.g., due to limits of the training corpus). Given individual mode recognizers with known accuracies, Wu, et al. (1999) describe a method for estimating a multimodal system's upper and lower performance bounds.

[10] Discriminative training is a training approach that directly reduces the recognition error instead of minimizing the mean-squared error between the model output and the training targets (Juang & Katagiri, 1992; Katagiri & McDermott, in press).

and normalization of recognizer output from speech and gesture to the same scale (see details in: Wu, Oviatt, & Cohen, 1999).

4. DIVERSITY OF EMERGING SPEECH AND GESTURE APPLICATIONS

Multimodal systems that recognize speech and pen-based gestures represent a very new field. This type of multimodal system first was designed and studied in the early 1990s, with the original QuickSet system prototype (see section 4.1) built in 1994. In this section, we describe four research-level systems that process users' speech and pen-based gestural input, as well as one example of a prototype (in section 4.3) that processes speech combined with 3D manual gesturing. These systems represent a variety of platforms, and they illustrate the diverse and challenging nature of emerging multimodal applications. With respect to functionality, the Human-centric Word Processor, Field Medic Information System, and Portable Voice Assistant (see section 4.2, 4.4 and 4.5) all integrate spoken words with pen-based deictic pointing events (i.e., selection of words or graphic objects), whereas both the QuickSet and VR Aircraft Maintenance Training systems (sections 4.1 and 4.3) process speech combined with varied gestures. With the exception of the Field Medic Information System, which supports alternative recognition of one input mode at a time, all of these multimodal systems time stamp the parallel speech and gesture input streams and then jointly interpret them based on a late semantic fusion approach. In most cases signal fusion is performed using a frame-based method, although QuickSet relies on a statistically-ranked unification process that functions within a hybrid symbolic/statistical architecture. Figure 4 summarizes the various types of multimodal speech and gesture systems that will be described in greater detail in this section, as well as their main functionality and architectural features.

[INSERT FIGURE 4 HERE]

4.1. OGI 's QuickSet System

QuickSet is an agent-based, collaborative multimodal system that enables a user to create and position entities on a map with speech, pen-based gestures, and direct manipulation (Cohen et al., 1997). These entities then are used to initialize a simulation. The user can create entities by speaking their names and distinguishing characteristics, while simultaneously indicating their location or shape with an electronic pen. For example, a medical unit could be created at a specific location and orientation by saying "medical company facing this way <draws arrow>." The user also can control entities in a simulation, for example by saying "Jeep 23, follow this evacuation route <draws line>" while gesturing the exact route with the pen. The QuickSet interface is illustrated in Figure 5 running on a hand-held PC. In addition to multimodal input, commands can be given just using speech or gesture as individual input modalities.

[INSERT FIGURE 5 HERE]

When used together, speech and gesture input are interpreted by parallel recognizers, which generate a set of typed-feature-structure meaning-fragments for each input stream. QuickSet then is able to fuse this partial information from these two modes by unifying any temporally- and semantically-compatible meaning fragments (Johnston et al., 1997). QuickSet then ranks the final joint interpretations statistically (Wu, Oviatt, & Cohen, 1999), and selects the best joint interpretation from the n-best alternatives to confirm to the user. During this process, individual modes can disambiguate one another, which effectively suppresses errors (Oviatt, 1999a, 1999b). In this respect, QuickSet permits the strengths of each mode to assist concretely in overcoming weaknesses in the other mode.

QuickSet has been developed as a set of collaborating agents using the Open Agent Architecture as its infrastructure (Cohen et al., 1994; Martin, Cheyer, & Moran, 1999), as illustrated in Figure 6. Agents can be written in Java, C, C++, Prolog, Visual Basic, Common Lisp, and other languages, and can reside anywhere on the Internet. These agents communicate with a Horn Clause language through a centralized facilitator, which routes queries, responses, requests, etc., to agents that have advertised relevant capabilities. The facilitator also supports triggering, thereby enabling asynchronous communication.

[INSERT FIGURE 6 HERE]

Among the agents that comprise QuickSet's main components[11] are: (1) continuous speaker-independent speech recognition[12], which for different applications have used recognizers such as IBM's Voice Type Application Factory, Microsoft's Whisper, and Dragon Systems Naturally Speaking, (2) the Members-Teams-Committee gesture recognizer (Wu, Oviatt, & Cohen, 1999), (3) a multimodal natural language parser (Johnston, 1998), (4) a unification-based multimodal integration component (Johnston et al., 1997), and (5) a map-based user interface, which supports different styles of confirmation (McGee, Cohen, & Oviatt, 1998). Other agents shown in Figure 6 also can support text-to-speech output, bridges to other software integration frameworks (e.g., Corba and KQML), and so forth. Most of these agents can run stand-alone on a handheld PC, or they can be distributed over a network.

In virtue of QuickSet's centralized facilitator architecture, when a second user interface is connected into the system, QuickSet then becomes a collaborative application with the various

[11] Unless stated otherwise, the agents described here were written at OGI.

[12] For QuickSet's military simulation application, the speech vocabulary is approximately 660 words, and the gesture vocabulary is 190 gestures.

user interfaces reflecting the result of each user's work. With a simple modification to this architecture in which the "wizard" (i.e., research assistant) becomes an unseen collaborator, the system can be configured as a "wizard-of-Oz" data collection environment (Clow and Oviatt, 1998). The agent architecture also enables the system to scale from handheld to wallsize interfaces, and to operate across a number of platforms (e.g., PCs, Unix workstations) and operating systems (e.g., Windows CE, Windows 95/98 and NT, versions of Unix).

The QuickSet system has functioned favorably during user testing, in part because of the proactive user-centered empirical work that provided the foundation for its design (Oviatt, 1997; Oviatt, DeAngeli, & Kuhn 1997). During a recent case study involving a domain expert (i.e., US Marine Corps major) who was engaged in an exercise initialization task, a 9-fold reduction in entity creation time was demonstrated while he interacted multimodally using QuickSet, in comparison with using the standard graphical interface for the same task (Cohen et al., 1998). QuickSet has been used for several map-based and VR applications, including military simulations, training systems, 3-D virtual-terrain visualization systems, a community disaster management system, a medical informatics system, and so forth (Cohen et al., 1997; Cohen et al., 1998; Cohen et al., 1999; Oviatt, 1999a). QuickSet also has been transferred to the research laboratories of each of the US armed services, and to many other government, commercial, and university sites, where it is being integrated with other software.

The componential nature of QuickSets agent architecture has resulted in relatively easy reuse and integration of QuickSets agents with other systems. For example, the QuickSet interface currently is being applied to a mobile augmented reality application (Feiner et al., 1997), and to the Naval Research Laboratorys (NRL) 3D Dragon 2 virtual-reality system (Cohen et al., 1999). In the latter interface, the user can speak while gesturing with a six degree-of-freedom "flight stick" that draws digital ink on a 3D terrain. Additional research using QuickSet currently is examining multimodal interaction in virtual worlds, in mobile environments (Oviatt, 2000), and also multimodal interfaces that support flexible interaction with tangible everyday objects (McGee, Cohen, & Wu, 2000).

4.2. IBM 's Human-Centric Word Processor

The Human-Centric Word Processor (HCWP) grew out of the MedSpeak/Radiology project (Lai & Vergo, 1997). MedSpeak is a continuous, real-time speech recognition system that accepts medical dictation, and uses natural language understanding to control the application. In comparison, HCWP combines speech recognition and natural language understanding (Papineni, Roukos, & Ward 1997) with pen-based pointing and selection gestures to create a multimodal word processing system. It is designed to solve one of the main usability problems of typical speech dictation systems— the need to correct, manipulate, and format text in a facile manner *after* it has been dictated (Vergo, 1998). This need for post-dictation corrections is motivated by the fact that: (1) people do not always dictate well-organized, grammatically correct text, (2) speech recognition systems are imperfect, so not all words are transcribed correctly, and (3) most people find it easier to edit and format their text *after*, rather than during, dictation. The HCWP system aims to support error correction and post-dictation manipulation of text by permitting flexible multimodal interaction using spoken language, an electronic stylus, and natural language processing.[13]

Text editing is basically a spatial task, which requires manipulation and movement of text elements. The vast majority of users, or nearly 100%, prefer to interact multimodally when

[13] A multimodal dictation application with error correction capabilities also has been developed by the Interactive Systems Laboratories at CMU (Suhm, 1998).

functioning in a visual-spatial domain. Some typical examples of multimodal constructions handled by HCWP include:

Example 1: "Delete this word <points to word>."

Example 2: "Change this date to the third <points to date>."

Example 3: "Underline from here to there <points to start and end of text line>."

Example 4: "Move this sentence here <points to sentence and new location>."

Examples 1 and 2 are accompanied by a single pointing event for selection, whereas examples 3 and 4 are accompanied by two pointing events to designate a text line or text movement. In example 2, the user only needs to point in the vicinity of a date for the system to understand which text elements are involved. Spoken language also can be used to specify the scope of an operation, as in "Delete this sentence <points to vicinity>" or "Delete this paragraph <points to vicinity>." Example 4 in particular illustrates the power of multimodal interaction in the text editing domain, since executing this same command with the standard GUI entails selecting the source text, cutting it, identifying the destination and, finally, pasting it in place. In all four examples, the ability to express these requests multimodally results in reduced task completion time, compared with either traditional GUI techniques or a speech-only dictation system.

In HCWP, the speech recognition vocabulary size is approximately 64,000 words, which is available for commands and dictation but primarily utilized during dictation mode. With this large vocabulary, the user can phrase action requests in an unconstrained manner, in contrast with a dynamic vocabulary or grammar-based approach to speech recognition. The interpretation of multimodal input to HCWP uses a statistical engine to interpret the natural language and gestural input, which was based on wizard-of-Oz experiments (Vergo, 1998).

All systems like HCWP that support the use of speech for both dictation and commands face the problem of categorizing each utterance correctly as dictated text or as a command. Some systems employ a “modal” approach with the application either in dictation or command mode. Mode switching is usually accomplished via a GUI button or verbal command. Some modern systems are modeless, with the system determining whether an utterance is a command or dictated text. Although the modeless approach is the holy grail, when the system makes a mode error this approach can lead to spiral errors and other severe usability problems (Karat et al., 1999). HCWP uses a two-position push-to-talk rocker switch on the microphone, so that the user can intentionally select to dictate text or issue a command to the recognizer. Observation has indicated that users rapidly adapt to the use of the rocker switch, and mode errors are minimal using this method.

The HCWP system also uses an LCD tablet with a stylus for gestural input. A pen stylus is a natural, precise, and effective mode of interaction for spatially-oriented tasks (Oviatt, DeAngeli, & Kuhn, 1997; Wolf & Morrel-Samuels, 1987). However, at present the only gesturing supported by the system is pointing. These deictic gestures are detected as asynchronous events, and are stored in the deictic gesture history along with context-dependent information, as illustrated in Figure 7.

[INSERT FIGURE 7 HERE]

The HCWP system maintains a notion of the context of the conversation, which can change as a result of any input modality specified by the system designer. A set of heuristics was developed to govern the determination of system context, with the following rules applied in priority order:

1. If there are pointing events associated with the spoken command, use them to determine the object of the command
2. If there is selected text on the screen, it has the highest priority
3. If there is no selected or otherwise indicated text, then the word(s) at the current cursor position are the object of the command
4. When a command is given unimodally using speech (e.g., "Underline the date") then, following the execution of the action, the cursor is left at the end of the affected string
5. After a command is executed, any text selections are unselected

The above rules permit the following natural sequences of interaction to be interpreted correctly:

- "Underline this sentence <pointing>. Make it red."   (uses principle 1, 5 and 3)
- <select some text> "Make it red."   (uses principle 2)
- "Move this sentence over here <points to sentence and new location>. Underline it."   (uses principle 1, 5 and 3)
- <select text ABC> "Underline it. Underline EFG."   (uses principles 2 and 5)

During spoken language processing, any pointing events are considered a "feature" that the statistical translator takes into account when evaluating conditional probabilities associated with formal language statements. Basically, the NLU engine receives information about whether and how many pointing events were associated with the natural language statement, and the NLU engine responds by selecting an appropriate formal language statement. Time-stamped pointing events stored in a deictic history buffer then are aligned with the time-stamped sub elements of formal language statements. Formal language statements produced by the NLU engine are flat text representations, which are transformed by the parser into top-level NLUEvent objects that reflect the language's domain/task dependent structure. The parser then sends NLUEvents to one of the application agents, which carries out the corresponding actions. The dispatcher decides where to send NLUEvents based on the event, its dialogue context, and the application's input focus, as summarized in Figure 7.

4.3. Boeing 's Virtual Reality Aircraft Maintenance Training Prototype

Boeing's Virtual Reality (VR) Aircraft Maintenance Training prototype is intended for use in assessing the maintainability of new aircraft designs and training mechanics in maintenance procedures using virtual reality (Duncan et al., 1999). A large percentage of the life cycle cost of an airplane is associated with maintenance, so making airplanes cheaper to maintain is a high priority. Since VR interfaces permit a user to perceive and interact with 3D objects directly, mechanics can use such interfaces to "walk through" and assess the maintainability of an aircraft design in the planning stages. Once a maintainable and cost-effective design is established, VR interfaces also can provide a simulated training environment for mechanics to both learn and practice procedures without taking an actual aircraft out of service.

The model scenes shown in Figure 8 illustrate the prototype VR maintenance training system. Figure 8 represents the main equipment center beneath the cockpit of a Boeing 777 airplane. The system prototype features an avatar driven by magnetic trackers attached to a human actor, so that the avatar's motions in the virtual environment shadow a human's motions in the real physical environment. The prototype task involves replacing a supplemental cooling check valve behind the P210 power maintenance panel in the VR scene.

[INSERT FIGURE 8 HERE]

Current VR interfaces are gesture-based and tend to be unnatural, frustrating to learn, and generally difficult to use. Besides the lack of haptics in most VR interfaces, the major shortcoming of prevailing gesture-based VR interfaces is that they fail to utilize the power of speech, or to accommodate the fact that human communication is multimodal. As an alternative to the standard approach, Boeing's system employs speech understanding and generation as part of the VR interface. When working in the VR environment, the user can decide when to gesture and when to speak and can use these modes alone or in combination. For example, the user can point to an object and say, "Give me that." Alternatively, if the object is distant, occluded, or otherwise out of view she might say, "Hand me the socket wrench." In another case, the user might say, "Take me to the E4 table rack" to fly to that location. Once there, she can physically walk slightly to the left or right to position her avatar body more precisely, use a flying gesture, or simply say, "Fly forward " to reposition herself.

To handle speech recognition, the system uses the IBM ViaVoice98 speaker-independent large-vocabulary speech recognizer and integrates its results with recognition of manual gestures. A high-quality microphone is used in open-dictation mode to capture the user's speech, and natural language processing is initiated when the keyword "over" is spoken. The system uses a Cyberglove gesture input device from Virtual Technologies and the GesturePlus 3D-gesture recognizer, which was programmed to recognize seven gestures. The graphics rendering is done using the Division system, which handles inverse kinematics, detailed collision detection, and interprocess communication. As an example of gesture recognition, the user can hold her palm flat with fingers extended to request a "flying" action through the VR environment. Since the Cyberglove is not nimble at fine-motor tasks, the user can select speech for tasks like removing small caps.

The natural language technology used to understand spoken commands is the same that supports other Boeing NLU applications, in particular its message processing (Duncan et al., 1994; Nicolino, 1994) and grammar checking applications (Homback, Duncan & Harrison, 2000; Wojcik & Holmback, 1996). As illustrated in Figure 9, it includes a syntactic parser and grammar interleaved with a semantic interpretation module. The output for each sentence is a graph representing the word sense and semantic relations between words and phrases. This sentence-level representation is integrated into a discourse representation using reference resolution algorithms and other discourse-level processing. The discourse representation consists of frame representations of the entities and events in the discourse. Although this NLU technology originally was developed to handle text-processing applications, it is being adapted and extended for use in several types of multimodal speech/gesture interfaces beyond the VR training application described here.

[INSERT FIGURE 9 HERE]

As illustrated in Figure 10, while the NLU subsystem interprets speech input using linguistic and domain knowledge, the gesture system simultaneously interprets gestures from the Division virtual reality systems “gesture actor.” Although speech is the primary driver of language interpretation, with gestural information used to fill in slots in the speech event frame (e.g., object identification, location), gesture input alone also can drive the downstream application. When the system’s temporal constraints are met, the integration subsystem combines time-stamped spoken language and gestural frames. If the NLU frames contain information like a deictic term, further information is sought from the gesture frames about the location and/or identity of the object indicated in the virtual world. The integration subsystem then sends the integrated frames to the command generation subsystem, which builds final commands that the system's "visual actor" can render.

[INSERT FIGURE 10 HERE]

The system configuration depicted here is realized as a distributed architecture using heterogeneous platforms (e.g., SGI Onyx, PC's, special-purpose hardware) and languages (e.g., Lisp, C++, Visual Basic), with components communicating via TCP/IP. To implement the VR elements, Division's immersive dvMockup software[14] was used, along with a head-mounted display and Ascension's Motion Star body tracking system.

Compared with a standard gesture-based VR interface, this prototype provides a more satisfactory way for users to engage in natural command and control in a VR world. Future work on the VR training application will extend the present prototypes limited domain functionality, add more discourse capabilities as well as natural language generation, incorporate earlier error repair, add a stylus, test different speech recognizers, and conduct formal usability testing with the second system prototype now under construction. The same basic architecture also is being adapted for use in an application involving human-robot cooperation, which potentially could be used to support future space station work. This will involve modifying the architecture depicted in Figure 10 to expand the integration component into a full discourse integration system that will implement a model of agent communication semantics (see, for example, Holmback, Greaves, & Bradshaw, 1999). For this purpose, a full natural language generation system (NLG) and speech synthesizer are being added, which will facilitate more complex spoken interaction between the user and system. In addition, the VR components will be replaced with the vision and gesture components of a mobile robotic platform. One overriding goal of our system's design is to support sufficient modularity that the NLU, NLG, and discourse integration components can be integrated with other future applications that require a multimodal interface.

4.4. NCR 's Field Medic Information System

The Field Medic Information System prototype was developed by NCR Corporation in collaboration with the Trauma Care Information Management System Consortium (Holzman, 1999). The system permits medical personnel (e.g., ambulance crews) to document patient care as it occurs in the field, including the entry of information about means of injury, patient assessment, treatment, triage and evacuation priority, and patient profile (e.g., identification, age, gender). This information then is forwarded electronically to a hospital in preparation for the patient's arrival. The system was designed to address field medics' current difficulty using paper forms to document patient care rapidly and accurately, especially under emergency circumstances when their eyes and hands must be devoted to assessing and treating the patient. More than 100 members of the medical community participated in knowledge engineering and system evaluation sessions that led to the current Field Medic Information System prototype. The system is comprised of two major hardware components— the Field Medic Associate (FMA) and the Field Medic Coordinator (FMC).

[INSERT FIGURE 11 HERE.]

The FMA is a flexible computer worn around the waist or in a vest, which uses a headset with earphones and a noise-reducing microphone for entering speech, as illustrated in the bottom of

[14] This software runs multiple sessions across the network and permits display of a "third-person" point of view on the VR scene.

Figure 11. As the medic speaks into the microphone, the FMA recognizes this input as data associated with specific fields in an electronic patient record. When the medic speaks to the patient record, the system confirms with a "ping" sound as audio feedback whenever the recognizer judges that speech input is consistent with its vocabulary and syntax, but with a "click" if not recognized as a valid entry. The user also can query the electronic record, for example by saying "Play back record," which produces synthesized speech readout of the record. In the future, the FMA 's interface will permit querying and editing specific fields. For example, the medic will be able to request " Play back triage priority," and then say " Delete" to edit that part of the record.

The FMA uses a speaker-independent speech recognition system. Its vocabulary and grammar were derived from several medical sources, including words and phrases appearing on medical forms, spontaneous spoken reports made by medical personnel during field training exercises and actual medical incidents, and extensive interviews with military and civilian medical personnel. In addition, potential users field-tested the system with medical scenarios, and the system was iterated to accommodate any words or phrases that were not already known. The resulting system has a 425-word vocabulary and the ability to accommodate approximately 8,700 phrases. Many of these phrases are alternative means of expressing the same information (e.g., "abrasion left lower arm" and "abrasion lower left arm"), resulting in the same input to the electronic patient record. Civilian and military medical users reviewed the vocabulary and grammar, as well as sample scenario scripts that illustrated their application. They judged the vocabulary and grammar to be sufficiently large and flexible to permit them to complete a field patient form orally for virtually any emergency medical situation, and with little or no training on the speech system. The next step will be to field test the system during actual medical incidents.

The current FMA speech recognition system does not permit entry of the patient's name, which would be out of vocabulary, nor free-form entry of descriptive information such as patient complaints. However, it supports patient identification via numbers (e.g., Social Security or other ID number), and both the FMA and FMC can read identification and medical history from a smart card carried by the patient. The FMA architecture also accommodates digital recording, but not recognition of free-form input.

[INSERT FIGURE 12 HERE.]

The Field Medic Coordinator (FMC) is a hand-held tablet computer. It displays the patient record illustrated in Figure 12, and permits the medic to modify it, either through the same speech-audio user interface incorporated in the FMA or by a series of quick pen taps to select items on lists that describe the patient's condition and treatment. Medics also can make free-form notations on the record using pen input to transmit electronic ink. Alternatively, they can quickly create speech annotations, which are associated with specific fields or graphics of the human body, by simply speaking while holding their pen on that area (e.g., tapping on neck graphic while speaking a description of the injury). As confirmation, a tick mark appears in the location of the patient record that has received a voice annotation. Hospital personnel to whom the record is forwarded then can tap on that mark to hear the annotation playback. The FMC also has a touch screen with a pop-up keyboard for typing information (e.g., patient name and complaints) into fields that otherwise can't be completed via selection from predefined lists.

The FMC can share data with the FMA via a wireless local area network (LAN) that operates over distances of hundreds of meters, and it also can receive data over the LAN from patient physiological monitors. A single FMC can simultaneously track patient data transmitted from multiple FMAs (thus the term Field Medic "Coordinator"), and it can relay those records to a hospital via cellular telephone, satellite, or radio, while simultaneously maintaining availability for local field use.

Typically, medics will use speech input with the FMA while initially assessing and stabilizing a patient, at a point when they need to keep their eyes and hands free. Once the patient is stabilized and on the ambulance, medics then have more time to work with the pen-based visual interface of the FMC. Like the FMA, the FMC is not designed to recognize speech and pen input signals simultaneously, nor does it entail a fusion-based architecture. Rather, the speech and pen input modes are provided as *alternative* input modes in this application, with spoken input recognized by the FMA, and either speech or pen selection recognized by the FMC. Both speech and pen input are particularly compatible with the mobile field use needed for this application domain.

4.5. BBN's Portable Voice Assistant

The Portable Voice Assistant (PVA) is a pen/voice multimodal interface that enables the user to choose the most efficient input mode for accessing and entering data on the World Wide Web (Bers, Miller, & Makhoul, 1998). It is available over a wireless network using a handheld device for the user to browse catalogues, order items, and complete HTML forms. The first prototype application developed using the PVA architecture is an on-line vehicle repair manual and parts ordering system, for which the intended users are army personnel repairing field equipment. For mobile applications such as this, in which the user 's visual attention and hands are occupied, speech input and output is particularly attractive.

Figure 13 illustrates the PVA interface, with the web page divided into two frames. The small upper frame has GUI buttons for controlling the speech recognizer, and it displays the system's status. The lower frame displays diagrams from the parts catalog or order form. The PVA uses

BBN's Hark speech recognizer, which is a speaker-independent continuous recognition system with an application vocabulary size of about 200 words. The default click-to-talk mode is well suited for noisy environments, although during quieter usage an open-microphone continuous-recognition mode permits the user to speak multiple commands without touching the screen. For handwriting recognition, Communications Intelligence Corporation's (CIC) software was used. The present system handles selection gestures on the graphic images, as well as handwriting recognition in the fields of the order form. The platform is a Fujitsu Stylistic 1200 pen-based PC, with Windows 95 and a RangeLan2 PC card for wireless network connectivity.

[INSERT FIGURE 13 HERE]

The PVA can interpret simultaneous pen/voice input so, for example, the user can say "Show me that part," while pointing at a screw on the display. To ensure proper synchronization, the PVA time stamps both speech and pen input events, and integrates them to form a frame-based description of the user's request, as illustrated in Figure 14. The speech thread generates events in response to output from the speech recognizer, and the pen thread generates selection events that are stored in a time-sorted buffer where they can be retrieved by the integrator thread. For the above example, the integrator simply would look for a pen event that occurred closest in time to the spoken deictic event "that". This design allows for asynchronous processing of spoken and pen-based input, while still processing them as coordinated multimodal pieces.

[INSERT FIGURE 14 HERE]

The PVA also can resolve references to an object via pen or speech input alone. To eliminate ambiguity, each of the approximately 50 parts known to the system has a unique ID, and objects are tagged in the parts diagrams when the application is configured. All words referring to parts are similarly tagged with this ID in the speech grammar. At runtime, this ID is included in the speech and pen input events when the object's name is spoken or its image is selected by the user's pen. This common representation permits unambiguous reference resolution through either modality. One advantage of this arrangement, for example, is that if the user speaks an out-of-vocabulary part name and system recognition fails, then she can recover by simply pointing at the part with her pen.

In a future version of the present prototype designed for general deployment, the speech vocabulary size ideally would be increased to handle more than 50 part names, and also to cover more variability in how real users refer to these parts. Vocabulary size should be guided in the future by user data collected during simulation and system testing. Although the size of the current prototypes vocabulary is modest, the basic client-server architecture used for distributing speech recognition (see Figure 15) permits scaling to very large vocabularies.

[INSERT FIGURE 15 HERE]

Our experience using the PVA indicates that a visually oriented web-browsing and ordering application is very well suited to a multimodal interface. In addition, multimodal interaction is optimal during dynamically changing mobile environments, because the user can flexibly adapt her interaction in response to task and environmental demands. Future field applications of this system could include point-of-sale purchase records, medical diagnostics, and in-car navigation systems.

4.6. Limitations of Current Speech and Gesture Multimodal Systems

All of the multimodal speech and gesture systems outlined above have been built since the mid-1990s, and they still are research-level systems. However, in some cases they have developed well beyond the prototype stage and are beginning to be integrated with a variety of other software at both academic and federal sites (Cohen et al., 1999). Therefore, although the field is developing rapidly, there are not yet commercially available systems of this type. To reach this goal, more general, robust, and scalable multimodal architectures will be needed, which are just now beginning to emerge. In addition, substantially more evaluation will be needed to guide the iterative development and optimization of these systems.

In addition, although it is well known that users have a strong preference to interact multimodally, and multimodal interaction offers many performance advantages that have been outlined in section 2, nonetheless not all system design is necessarily best approached with a multimodal interface. Empirical work has documented that the largest performance gains for multimodal pen/voice systems occur in visual/spatial domains (Oviatt et al, 1997). If an application is developed that has little or no spatial component, then it is far less likely that users will communicate multimodally. In such a case, a unimodal interface may be appropriate. Finally, although a multimodal pen/voice combination is an attractive interface choice for next-generation systems due to the mobility, transparency, and expressive power of these particular input modes, nonetheless other modality combinations also need to be explored, and will be preferred by users for certain applications. Section 5 presents several of the major research directions that will need further work before new multimodal systems can be developed fully and eventually commercialized.

5. FUTURE RESEARCH DIRECTIONS FOR MULTIMODAL INTERFACES

Advancing the state of the art of multimodal speech and gesture systems has depended on hardware advances in new media, the construction of new concepts for multimodal prototype systems, substantial empirically-oriented research with human subjects, and the development of appropriate metrics and techniques for evaluating alternative multimodal system designs. However, to develop successful and varied multimodal systems of the future, ones with better performance characteristics than unimodal interfaces or GUIs, many fundamental scientific issues and multidisciplinary research challenges remain to be addressed. In this section, we discuss several of these key research challenges.

5.1. Cognitive Theory and Empirical Science Underpinnings

First, a better understanding will be required of the unique linguistic and performance characteristics of natural communication modalities, such as human speech, gesture, gaze patterns, and facial expressions. Related cognitive science literature is available on the performance characteristics and organization of each of these human systems from psychology, linguistics, neuroscience, and elsewhere. Psychological research also has provided relevant empirical results and theoretical grounding on how humans deploy their attention and coordinate two or more modes during the execution of complex tasks, especially in cases where one mode is visual and the other auditory (see Wickens et al., 1983, 1984, on “Multiple Resource Theory”). With respect to speech and gesture, both linguistic and psychological literatures also have examined how people coordinate their spoken language and natural manual gesturing during interpersonal communication (Kendon, 1980; Levelt, 1985; McNeill, 1992).

In contrast, limited literature is available on how different communication modalities or combined modes are organized specifically during human-computer interaction. For example, only recently have detailed empirical results been summarized on how people integrate and synchronize speech and pen input during human-computer interaction (Oviatt, DeAngeli, & Kuhn, 1997). These findings were critical in establishing the temporal constraints and other key architectural features for building the QuickSet multimodal system (described in section 4.1). Recent work also has revealed many unique linguistic features of multimodal language, compared with spoken or keyboard interaction (Oviatt & Kuhn, 1998; Oviatt, 1999b), and this information has been used to establish basic natural language processing techniques specifically for multimodal systems.

In the future, the relevant cognitive science literatures should be utilized more extensively as a basis for: (1) hypothesizing about and examining multimodal human-computer communication, (2) spearheading parallel empirical work on other promising modality combinations besides speech and gesture (e.g., speech and gaze tracking), and (3) proposing innovative new system designs that are consistent with human information and sensory processing advantages. Further empirical work will be needed to generate the necessary foundation of predictive information for guiding the design of new multimodal systems. Finally, to advance multimodal systems, further cognitive and neuroscience theory will be needed that specifically addresses issues such as: (1) the behavioral features and automaticity of human communication modes, (2) how the perception and production of one mode is altered by the presence of a second mode (as in the seminal work by McGurk & MacDonald, 1976), and (3) basic inter-modal coordination and synchronization issues.

5.2 New Multimodal Interface Concepts

Although the main multimodal subliteratures have focussed on either speech or pen input and speech and lip movements (Stork and Hennecke, 1995; Rubin, Vatikiotis-Bateson, and Benoit, 1998), recognition of other human input modes also is beginning to mature and be integrated into new types of multimodal systems. In particular, there is growing interest in designing multimodal interfaces that incorporate vision-based technologies, such as tracking and interpretation of gaze, head position, body location and posture, facial expressions, and manual gesturing. These kinds of vision-based technology, recently referred to as "perceptual user interfaces" (Turk and Robertson, 2000), unobtrusively monitor user behavior. That is, they involve *passive* human input that requires *no explicit user command to the computer at all*. This contrasts with *active* input modes, such as speech, pen-based gestures, or other manual input, which the user *intends as a command issued to the system*. While passive modes may be less obtrusive, active modes generally are more reliable indicators of user intent.

As vision-based technology and perceptual interfaces mature, one future direction for multimodal interface design is the development of a *blended interface style* that combines both a passive and active mode. A blended multimodal interface can be temporally cascaded such that advance information arriving from the passively-tracked mode (e.g., eye gaze) is used to improve the multimodal system's prediction and interpretation of the active mode that follows (e.g., manual or speech input). An example of a cascaded passive/active interface is the IBM MAGIC system. MAGIC passively tracks a user's gaze at a text field (i.e., right, left, above or below the cursor location) and uses this information to predict the direction of cursor movement and to modulate a manual track pointer's physical resistance (Zhai et al, 1999). This particular multimodal interface aims to decrease the user's manual fatigue and increase input efficiency.

This type of hybrid multimodal interface potentially can perform more reliably than a pure passive-tracking system, because the active input mode is available to clarify ambiguous user intentions. Early information from the passive mode also can provide predictive power that enhances system robustness and delivers usability advantages, compared with an active mode alone. In the future, this kind of blended multimodal interface may provide the user with greater transparency, better control, and an improved usability experience, while also supporting broader application functionality.

Finally, as new types of multimodal systems proliferate in the future, they also are increasingly likely to include more than two input modes. This trend already has been initiated within the field of biometrics research, which has combined multiple behavioral modes (e.g., voice recognition, handwriting recognition) with physiological ones (e.g., fingerprints, retinal scans) using sensor fusion technologies (Pankanti, Bolle, and Jain, 2000). The driving goal behind this trend to add modes has been improvement of the reliability of person identification and verification tasks within a wider range of realistic usage conditions. As research progresses in the demanding area of biometric security applications, future work will be needed to transfer the more promising new techniques and architectures to other types of interactive multimodal systems.

5.3 Multimodal Language and Dialogue Processing

To provide a scientific foundation for developing multimodal speech and gesture dialogue systems, a general theory of conversational interaction will be needed, as will relatively media-independent representations of intent, semantic content, and context. The generally accepted media-independent level for formulating the basic principles governing dialogue interaction is *communicative intent,* which often is represented in terms of speech acts (Searle, 1969). The same underlying speech act representations are used in dialogue systems for recognition and production, using techniques from planning (Appelt, 1985; Cohen & Perrault, 1979), plan recognition (Allen & Perrault, 1980; Carberry, 1990), formal logic and automated reasoning (Cohen & Levesque, 1990; Perrault, 1990; Sadek, 1991), and statistical language modeling (Reithinger & Klesen, 1996). Although general purpose representations and theories have been available for a number of years, only recently has a real-time speech-only dialogue system been developed that operates at the level of the speech act (Bretier & Sadek, 1997), and for which the basic representations and processing techniques are common between input and output. Regarding multimodal output, the same types of speech act representations and planning processes have been developed to coordinate the presentation of text and graphics (Feiner & McKeown, 1991; Wahlster et al., 1993). Future research will need to address the development of complete *multimodal* dialogue systems that incorporate a level of intent representation suitable for non-speech modalities such as gesture. Furthermore, there needs to be agreement among researchers on a speech act vocabulary that is sufficient for fostering community-wide comparisons and incremental progress.

Regarding media-independent representations of the content of these speech acts, there has been de facto agreement on feature/frame structures, coupled with rules and temporal constraints, to represent multimodal input. Likewise, similar structures and procedures have been developed for planning and coordinating multimodal output (Feiner & McKeown, 1991). Future research will need to synthesize these approaches, employing common representations of temporal information (Allen, 1984). Furthermore, research is needed to develop general media-independent theories of information presentation, both cognitive and formal, that can be realized directly as architectural principles and functioning software. For example, unified theories will be critical in guiding animated character displays that involve nonverbal gestures and facial expressions as communicative output (Andre, Rist, & Muller, 1999; Cassell & Stone, 1999; Cassel et al., 2000; Lester et al., 1999).

The glue between input and output is the representation and use of context. Unlike graphical user interfaces that are designed to be context independent, systems based on natural language have long been designed to represent and use context, for example, to interpret pronouns. More generally, context will need to be derived from the user's prior *multimodal* interactions, and the system's multimedia presentations. The latter includes the visual context of what is on the screen and, for future systems that employ camera input, the user's physical context. Unfortunately, most multimedia systems simply present information without representing its context. Future systems that *synthesize* multimedia (e.g., Feiner & McKeown, 1991; Roth, Mattis, & Mesnard, 1991) will be needed to capture visual context adequately. Precisely what to save from prior multimodal interaction, and how to deploy these contextual representations to best ensure correct multimodal interpretations, is a crucial subject of future research.

Finally, to arrive at multimodal interpretations, future 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, together with statistical language modeling. Given this framework, a complete multimodal dialogue system could perhaps be trained on an annotated corpus of interactions that provides the basis for correlating speech, gesture, natural language, context (both linguistic and visual), task scenario, and user intent. To accomplish this, large-scale multimodal corpora need to be developed and annotated. In addition, statistical interpretation algorithms, such as MTC, will need to be extended and tested on these new corpora.

5.4 Error Handling Techniques

Fragile error handling currently remains the number one interface problem for recognition-based technologies like speech and pen (Karat, et al., 1999; Oviatt, 2000; Roe & Wilpon, 1994; Rhyne & Wolf, 1993). However, as discussed earlier in section 2, multimodal interface designs that combine two input modes tend to have superior error handling characteristics (Oviatt & vanGent, 1996; Oviatt, Bernard, & Levow, 1999; Oviatt, 1999a; Rudnicky & Hauptmann, 1992; Suhm, 1998; Suhm, Myers, & Waibel, 1996). Future research needs to continue to investigate strategies for developing graceful error handling in multimodal systems. In particular, research should explore issues such as: (1) methods for designing multimodal architectures that support higher levels of mutual disambiguation between signals, (2) the impact of incorporating a third input mode on error avoidance and system stability, (3) the impact of new language and dialogue processing techniques on error avoidance and resolution, and (4) adaptive architectures that reduce errors and stabilize system performance in noisy mobile environments and other challenging contexts. Eventually, improved error avoidance and correction techniques also need to become available to application developers in the form of toolkits (Vo, 1998).

5.5 Adaptive Multimodal Architectures

The future research agenda for developing adaptive multimodal speech and gesture architectures subsumes the problems of what and when to adapt, as well as how to adapt multimodal systems so that their robustness can be enhanced. Adaptive multimodal architectures will increase the ease with which users can interact with a system by continually adjusting to a user and her surroundings. Two primary candidates for system adaptation are *user-centered* and *environmental* parameters.

With respect to user-centered parameters, previous empirical work on multimodal pen/voice integration has revealed two main types of user— ones who habitually deliver speech and pen signals in an overlapped or simultaneous manner, and others who synchronize signals sequentially with pen input preceding speech by up to 4 seconds (Oviatt, 1999b). Any given user 's habitual integration pattern is apparent at the beginning of their system interaction. As a result, future multimodal systems that are capable of distinguishing these two types of user, and adjusting the system's temporal thresholds for integrating signals accordingly, potentially could achieve greater recognition accuracy and interactive speed.

With respect to environmental parameters for adapting multimodal pen/voice systems, background noise and speech signal-to-noise ratio (SNR) are two widely used audio dimensions that have an impact on system recognition rates. To design robust mobile pen/voice systems for field use, future research will need to experiment with adaptive processing that tracks the background noise level and dynamically adjusts the system's weightings for speech and gesture input from the less to more reliable input mode (Rogozan & Deleglise, 1998). In addition, future research needs to establish an analogue to SNR for calibrating noisy pen input and its impact on gesture recognition rates, which could be especially consequential during movement, fatigue, or device-induced jitter. In general, the quality of speech and pen input signals can be estimated by relative recognition uncertainty, which some system recognizers produce along with posterior probabilities. The MTC approach outlined in section 3.4 estimates signal-level uncertainty, and potentially could be used to provide a base architecture for developing successful adaptive processing.

Most existing adaptive multimodal architectures were developed for audio-visual recognition of speech and lip movements. These architectures have relied on two general types of models, ones that use *parameter-based adaptive weights*, and others based on *black-box adaptivity*. For models based on parameter-based weightings, the system calibrates an individual modality's weighting (e.g., speech) as directly proportional to the magnitude of the relevant environmental parameter (e.g., strength of signal-to-noise ratio (Meier, Hurst, & Duchnowski, 1996), or probability dispersion weights (Adjoudani & Benoit, 1996)). That is, if SNR is a low value, then the reliability of the speech signal interpretation would be regarded as weak and this mode would be relatively underweighted compared with gesture input. In addition, different types of multimodal commands can be assigned different weights, just as modalities can (Wu, Oviatt, & Cohen, 1999). For black-box models (e.g. neural networks), the relation between variables and modality weights is built through supervised learning (Sejnowski et al., 1990; Waibel et al., 1995). This learnt relation is completely dependent on training data, and may be highly nonlinear. A major area in need of further investigation is the application of these adaptive processing methods specifically to pen/voice multimodal input.

5.6 Multi-Device Multi-User Ubiquitous Computing

Most of today’s multimodal applications have been built around a single device or closely coupled complex of devices. As we move towards an era of ubiquitous computing, people will find themselves in environments where they are not dealing with “a computer” per se. Rather, they will be moving among a variety of complementary devices, each appropriate for a particular task. For example, in an “interactive room” (Coen, 1998; Pentland, 1996) there may be display devices of all sizes (e.g., palmtop to wall-screen) and orientations (e.g., wall, table). They could be controlled with pointing devices that are direct or indirect (e.g., pen vs. trackball), in contact or distant (e.g., pen vs. laser pointer), or specialized for different kinds of manipulation (e.g., gloves). Cameras and microphones will create a *perceptual space*, in which what people say and do is gathered as input signals to be interpreted within the environment ’s overall computing context.

In future interactive environments such as this, multimodal interaction styles will predominate. One important future research goal will be to understand the nature of multimodal interaction within such a computing context. Another will be to create scalable and reusable architectures that cope effectively with the diversity of these input modes and signals— each with its unique characteristics involving response latency, recognition certainty, adaptability to the user, and so forth. This will include developing standard ways to indicate multiple alternatives for an input signal, and easy configuration/reconfiguration of components for interpreting input based on multiple data sources. In comparison with this vision, today's multimodal speech and gesture systems (e.g., outlined in section 4) eventually will be viewed as just the first step in a longer-term evolution away from systems based on specialized code and devices.

As multimodal interaction transforms into larger physical spaces and ubiquitous forms, multiple users often will be interacting together— pointing to a shared display surface, gesturing simultaneously, and talking in a natural style that includes interruptions, disfluencies, overlapped speech, and so forth. To date, the research in computer-supported collaborative work involving issues like turn taking and dialogue control has dealt primarily with individual communication modes, usually textual in nature. Such work will need to be extended to handle natural *multimodal* interactions— including the technological issues, but also understanding how groups of people will communicate and coordinate when using combined speech and gesture to accomplish daily computing tasks. If multimodal spaces of this kind are to become commonly accepted, they also will have to minimize the amount and/or size of gear that a user wears or carries. This will require greater dependence on ambient environmental devices, such as wide-field cameras and directional microphone arrays (Flanagan et al., 1991). It also will require the development of methods for tracking and recognizing an individual user's contributions to the overall signal environment. Among other things, this will necessitate basic changes in the way that operating systems deal with input, since today’s systems generally implicitly assume that a particular device provides input from a single individual at a given moment. In sum, the prospect of future multi-device multi-user ubiquitous computing presents a wide array of long-term research challenges.

5.7 Multimodal Research Infrastructure

In order for the fledgling multimodal research community to develop high-performance multimodal systems and eventually commercialize them, considerable research will be needed to develop appropriate infrastructure, including: (1) semi-automatic simulation methods for empirical data collection and prototyping new systems, (2) automated tools for collecting and analyzing multimodal corpora during realistic usage contexts, (3) novel metrics for evaluating multimodal systems, and (4) automated corpus collection and analysis tools for iterating new multimodal systems in order to steadily expand and improve their performance. Before multimodal systems can proliferate, the community also will need: (5) software tools that support the rapid creation of next-generation multimodal applications.

Tools for multimodal system design, iteration, and corpus collection

Multimodal systems still are very new and hard to build. Among other things, advancing this research area will require proactive research and situated data collection in order to achieve high quality multimodal system design. More specifically, software support is needed for developing semi-automatic simulation techniques, which are the preferred method for designing multimodal prototypes for systems still in the planning stages (Oviatt et al., 1992; Oviatt, 1996; Cheyer, Julia, & Martin, 1998). In a simulation environment, the user believes that she is interacting with a fully functional system, while a trained programmer assistant actually provides simulated system responses from a remote location. Simulation software is designed to support a subject-paced rapid interaction with the simulated system. It also is designed to be rapidly adaptable, so that alternative interface designs and system features can be investigated fully and easily. Simulation-based research can help designers to evaluate critical performance tradeoffs and make decisions about a system's design, which is necessary for creating usable multimodal systems that have valuable functionality.

In the future, simulation tools will need to be developed that permit researchers to explore new input modes and interface designs appropriate for a wide spectrum of different multimodal systems. There also is a special need for longitudinal user studies, especially on multimodal interfaces that incorporate novel media— ones with which users have limited familiarity. In addition, since there is strong interest in designing multimodal systems for use in natural field environments and while users are mobile, simulation and other data collection tools are especially needed for supporting situated data collection in such contexts (e.g., in-vehicle and emergency medical applications (see Holzman, 1999)). New tools are just beginning to be developed for this purpose (Oviatt & Pothering, 1998).

In the area of spoken language processing, the availability of significant corpora of transcribed and annotated training data has been a critical resource enabling the remarkably rapid progress in spoken language understanding during the past decade (Cole et al., 1995). Unfortunately, the unavailability of common multimodal corpora currently is a problem that has been impeding progress on the development of new methods for multimodal language processing (i.e., including both linguistic and statistical techniques), as well as techniques for multimodal signal fusion and adaptive processing. More widespread availability of multimodal corpora also is critically needed to develop appropriate methods for evaluating and iterating the overall performance of new multimodal architectures. The collection and analysis of large multimodal corpora could be expedited greatly by the development of automated multimodal data loggers. A few such multimodal data loggers are beginning to appear for recording, searching, analyzing, and generating automatic summaries about the signal-level and linguistic features of users' multimodal utterances, as well as system recognition results on multimodal input (Clow & Oviatt, 1998; also see MITRE's logger at: http://www.mitre.org/research/logger/release/1.0/html/logger.html).

Tools for multimodal software development

Multimodal systems are more complex than unimodal ones, making both system design and implementation more difficult. Moreover, recognizers for various modalities can be difficult to acquire and to integrate, because toolkit-level support is limited at best, and integration of recognizers in new applications requires significant expertise. There presently are no software tools to assist application designers with building multimodal applications. Only recently have programming toolkits become available for recognizing natural unimodal input such as speech (Kempf, 1994; Sun, 1998; Huang et al., 1995). These new single-mode toolkits usually are intended for use by recognition technology specialists or advanced programmers. However, an example of an exception to this is the CSLUrp interactive builder for speech applications (Sutton & Cole, 1997), which aims to provide tools for non-specialists, although it is limited to speech-only transaction interfaces.

In the future, developers will need prototyping tools for designing multimodal interfaces that combine two or more natural modes, possibly in conjunction with conventional direct manipulation. These tools ideally should be multimodal themselves, as well as incorporating programming by demonstration (Cypher, 1993) so that designers with limited programming experience can prototype multimodal designs rapidly. For example, using programming by demonstration techniques, a designer potentially could specify speech and gesture input operations for a wizard-of-Oz simulation, which could enable the simulation system to infer the user's multimodal behavior during testing, thereby reducing the human wizard's workload significantly. Prototyping tools also should support the informal, iterative techniques that designers currently use in the early stages of design— from sketching visual interface elements using a system like SILK (Landay, 1996; Landay & Myers, 1995; Landay & Myers, 1996), to storyboarding, to high-fidelity wizard-of-Oz simulation techniques originally developed for spoken language (Kelley, 1984) and more recently for multimodal systems (Oviatt et al., 1992; Cheyer, Julia, & Martin, 1998). These multimodal prototyping tools also should give designers the flexibility to fill in design details as choices are made (Wagner, 1990), and to test the designs interactively. For example, interactive tools should permit the designer to demonstrate wizard-of-Oz sequences that simulate the intended behavior of the recognizer, as the SUEDE system is intended to do for spoken language (Chen, 1999). Together, software tools that support prototyping techniques such as interactive sketching, programming by demonstration, and high-fidelity simulations can be used to design multimodal systems far more rapidly, which will be necessary for this new generation of interfaces to proliferate.

6. CONCLUSION

Multimodal systems that process users' speech and pen-based gestural input have become a vital and expanding field, especially within the past 5-8 years, with demonstrated advances in a growing number of research and application areas. Among the benefits of multimodal interface design are general facilitation of the ease, flexibility, and power of computing, support for more challenging applications and forms of computing than in the past, the expanded use of computing while mobile and in natural field settings, and a potentially major increase in the accessibility of computers to a wider and more diverse range of users. In particular, since multimodal systems support relatively natural interaction without special training, they will make computing and information services available to field workers, business and service people, students, travelers and other mobile users, children, the elderly, permanently and temporarily disabled users, and computer-illiterate and casual users.

In this paper, we have summarized both the prevailing and newly emerging architectural approaches that are available for interpreting dual input signals in a robust manner— including early and late semantic fusion approaches, as well as a new hybrid symbolic/statistical architecture for processing pen/voice input. We also have described a diverse collection of state-of-the-art multimodal systems that are capable of processing users' spoken and gestural input— ranging from map-based and virtual reality systems for engaging in simulations and training, to field medic systems for mobile use in noisy environments, to web-based transactions and standard text-editing applications that will reshape daily computing tasks. We have indicated that many key research challenges remain to be addressed before successful multimodal systems can be realized fully. Among these challenges are the development of cognitive theories of multimodal interaction, as well as more effective natural language, dialogue processing, semantic integration, and error handling techniques. While the rapid maturation of spoken language technology has contributed directly to recent advances in multimodal systems, nonetheless further strides will be needed in other component technologies and hardware before existing multimodal systems can be diversified and optimized fully. In addition, new and more sophisticated architectures will be needed for handling media-independent representations, for designing more robust and adaptive multimodal systems, and for supporting multi-device multi-person use. Before this new class of systems can proliferate, multimodal toolkits also will be needed to support software development, as will simulation-based methods for corpus collection, analysis, & effective system iteration.

Finally, some of the requirements for advancing innovative multimodal systems are not intellectual ones— but rather social, political, and educational in nature. The development of state-of-the-art multimodal systems of the kind described in this paper also requires multidisciplinary expertise in a variety of areas, such as speech and hearing science, perception and graphics, linguistics, psychology, signal processing, pattern recognition, statistics, engineering, and computer science. The multidisciplinary nature of this research makes it unlikely that a single group can conduct meaningful research across the entire spectrum. As a result, collaborative research and "community building" among multimodal researchers will be critically needed to forge the necessary relations among those representing different component technologies and key disciplines. In addition to cross-fertilization of ideas and perspectives among these diverse groups, there also is a critical need for cross-training of students and junior researchers. Like spoken language systems, multimodal technology does not fit neatly into a traditional academic departmental framework. To make the appropriate educational opportunities and resources available to future students, new academic programs certainly will need to be formulated that encourage and reward researchers who successfully reach across the boundaries of their narrowly defined fields.

NOTES

Background. This article is based partly on discussions held at a CHI '99 Workshop meeting in Pittsburgh, PA. during May of 1999. The workshop, "Designing the User Interface for Pen and Speech Multimedia Applications," was organized by James A. Larson (Intel Architecture Labs), David Ferro (Unisys Natural Language Understanding group), and Sharon Oviatt (Computer Science Dept. at OGI). In addition to the present authors, this paper also benefited from workshop discussions with Adam Cheyer of SRI, Christopher Esposito of Boeing, Jennifer Mankoff of

Georgia Tech, Mark Newman of UC-Berkeley, Pernilla Qvarfordt of Linköping University in Sweden, Dan Venolia of Microsoft, and Timothy Miller of Brown University.

Acknowledgements. We would like to thank Dana Director, Gwen Brake and Bridget Adams for their expert assistance with graphics and manuscript preparation.

Support. The preparation of this article has been supported in part by Intel Corporation, by NSF grant IRI-9530666 and NSF Special Extension for Creativity (SEC) grant IIS-9530666 to the first author. Additional support has also been provided by DARPA contracts DABT63-95-C-007 and N66001-99-D-8503, as well as ONR grant N0014-95-1164, to the second author.

First Author Address. Sharon Oviatt, Center for Human-Computer Communication, Department of Computer Science and Engineering, Oregon Graduate Institute of Science & Technology, 20000 N.W. Walker Road, Beaverton, Oregon 97006. Email: oviatt@cse.ogi.edu URL: http://www.cse.ogi.edu/CHCC/

REFERENCES

Adjoudani, A., & Benoit, C. (1996). On the integration of auditory and visual parameters in an HMM-based ASR. In D. Stork, & M. Hennecke (Eds.), *Speechreading by humans and machines, NATO ASI Series, Series F: Computer and Systems Science* (pp. 461-473). Berlin: Springer Verlag.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence, 23(2),* 123-154.

Allen, J. F., & Perrault, C. R. (1980). Analyzing intention in dialogues. *Artificial Intelligence, 15(3),* 143-178.

Andre, E., Rist, T., & Muller, J. (1999). Employing AI methods to control the behavior of animated interface agents. *Applied Artificial Intelligence*, *13*, 415-448.

Appelt, D. (1985). *Planning English sentences.* Cambridge, U. K.: Cambridge University Press.

Benoit, C., Martin, J-C., Palachaud, C., Schomaker, L., & Suhm, B. (in press). Audio-visual and multimodal speech systems. In D. Gibbon & R. Moore (Eds.), *Handbook of Standards and Resources for Spoken Language Systems*. Norwell, MA: Kluwer.

Bers, J., Miller, S., & Makhoul, J. (1998). Designing conversational interfaces with multimodal interaction. *DARPA Workshop on Broadcast News Understanding Systems*, 319-321.

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. IEEE Press.

Bretier, P., & Sadek, D. (1997). A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction. *Intelligent Agents III: Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages (ATAL-96)*, 189-204. Springer Verlag.

Calder, J. (1987). Typed unification for natural language processing. In E. Klein & J. van Benthem (Eds.). *Categories, Polymorphisms, and Unification* (pp. 65-72). Center for Cognitive Science, University of Edinburgh.

Carberry, S. (1990). *Plan recognition in natural language dialogue.* ACL-MIT Press Series in Natural Language Processing, MIT Press.

Carpenter, R. (1992). *The logic of typed feature structures.* Cambridge, U. K.: Cambridge University Press.

Carpenter, R. (1990). Typed feature structures: Inheritance, (in)equality, and extensionality. *Proceedings of the ITK Workshop: Inheritance in Natural Language Processing*, 9-18. Tilburg: Institute for Language Technology and Artificial Intelligence, Tilburg University.

Cassell, J., & Stone, M. (1999). Living hand to mouth: Psychological theories about speech and gesture in interactive dialogue systems. *Working notes of the AAAI'99 Fall Symposium Series on Psychological Models of Communication in Collaborative Systems.* Menlo Park, CA: AAAI Press.

Cassell, J., Sullivan, J., Prevost, S., & Churchill, E. (Eds.) (2000). *Embodied conversational agents*. Cambridge, MA: MIT Press.

Cheyer, A., & Julia, L. (1995). Multimodal maps: An agent-based approach. *International Conference on Cooperative Multimodal Communication, (CMC'95),* 103-113. Eindhoven, The Netherlands.

Cheyer, A., Julia, L., & Martin, J. C., (1998). A unified framework for constructing multimodal experiments and applications. *Conference on Cooperative Multimodal Communication (CMC'98),* 63-69. Tilburg, The Netherlands.

Clow, J., & Oviatt, S. L. (1998). STAMP: A suite of tools for analyzing multimodal system processing. *Proceedings of the International Conference on Spoken Language Processing,* 2, 277-280. Sydney: ASSTA, Inc.

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 (CHI'92),* 329-334. New York: ACM Press.

Coen, M. (1998). Design principles for intelligent environments. *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98).* AAAI Press.

Cohen, P. R., Cheyer, A., Wang, M., & Baeg, S. C. (1994). An open agent architecture. *AAAI '94 Spring Symposium Series on Software Agents,* 1-8. AAAI Press. (Reprinted in Huhns and Singh (Eds.). (1997). *Readings in Agents* (pp. 197-204). Morgan Kaufmann Publishers, Inc.)

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 (CHI'89),* 227-234. ACM Press. (Reprinted in Maybury & Wahlster (Eds.) (1998). *Readings in Intelligent User Interfaces* (pp. 29-37). San Francisco: Morgan Kaufmann.)

Cohen, P. R., Johnston, M., McGee, D., Oviatt, S. L., Clow, J., & Smith, I. (1998). The efficiency of multimodal interaction: A case study. *Proceedings of the International Conference on Spoken Language Processing, 2*, 249-252. Sydney: ASSTA, Inc.

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*, 31-40. New York: ACM Press.

Cohen, P. R., & Levesque, H. J. (1990). Rational interaction as the basis for communication. In P. R. Cohen, J. Morgan, & M. E. Pollack (Eds.), *Intentions in Communication* (pp. 221-255). Cambridge, MA: SDF Benchmark Series, MIT Press.

Cohen, P. R., McGee, D., Oviatt, S., Wu, L., Clow, J., King, R., Julier, S., & Rosenblum, L. (1999). Multimodal interaction for 2D and 3D environments. *IEEE Computer Graphics and Applications. 19(4),*10-13. IEEE Press.

Cohen, P. R., & Oviatt, S. L. (1995). The role of voice input for human-machine communication. *Proceedings of the National Academy of Sciences, 92(22),* 9921-9927. Washington, D. C.: National Academy of Sciences Press.

Cohen, P. R., & Perrault, C. R. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science*, *3(3),* 177-212. Cognitive Science Society, Inc.

Cole, R., Hirschman, L., Atlas, L., Beckman, M., Biermann, A., Bush, M., Clements, M., Cohen, P., Garcia, O., Hanson, B., Hermansky, H., Levinson, S., McKeown, K., Morgan, N., Novick, D., Ostendorf, M., Oviatt, S., Price, P., Silverman, H., Spitz, J., Waibel, A., Weinstein, C., Zahorian, S., & Zue, V. (1995). The challenge of spoken language systems: Research directions for the nineties. *IEEE Transactions on Speech and Audio Processing, 3 (1)*, 1-21.

Cole, R., Mariani, J., Uszkoreit, Zaenen, A., & Zue, V. (Eds.). (1997). *Survey of the state of the art in human language technology*. Cambridge, MA.: Cambridge University Press. (Also status report to the European Commission and the National Science Foundation, May 1994).

Cypher, A. (Ed.). (1993). *Watch what I do: Programming by demonstration.* Cambridge, MA: MIT Press.

Duncan, L. et al. (1994). *Message processing and data extraction for the Real-Time Information Extraction System (RTIMS) project* (Boeing Technical Report BCSTECH-94-057).

Duncan, L., Brown, W., Esposito, C., Holmback, H., & Xue, P. (1999). *Enhancing virtual maintenance environments with speech understanding*. Boeing M&CT TechNet.

Elman, J.L. (1990). Finding Structure in Time, *Cognitive Science* 14, 179-211.

Erman, L. D., & Lesser, V. R. (1975). A multi-level organization for problem solving using many diverse sources of knowledge. *Proceedings of the 4th International Joint Conference on Artificial Intelligence,* 483-490. IJCAI Press.

Feiner, S. K., & McKeown, K. R. (1991). COMET: Generating Coordinated Multimedia Explanations. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'91),* 449-450. ACM Press.

Feiner, S., MacIntyre, B., T. Hollerer, T., & Webster, A., (1997). A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. *International Symposium on Wearable Computing (ISWC'97).* Cambridge, MA: CS Press.

Fell, H., Delta, H., Peterson, R., Ferrier, L, Mooraj, Z., & Valleau, M. (1994). Using the baby-babble-blanket for infants with motor problems. *Proceedings of the Conference on Assistive Technologies (ASSETS'94),* 77-84. Marina del Rey, CA.

(URL http://www.acm.org/sigcaph/assets/assets98/assets98index.html)

Flanagan, J., Berkeley, D., Elko, G., West, J., & Sondhi, M. (1991). Auto-directive microphone systems. *Acustica*, *73*, 58-71.

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

Goldschen, A., & Loehr, D. (1999). The Role of the DARPA communicator architecture as a human computer interface for distributed simulations*.* *Spring Simulation Interoperability Workshop*. Orlando, Florida: Simulation Interoperability Standards Organization.

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

Hauptmann, A. G. (1989). Speech and gestures for graphic image manipulation. *Proceedings of the International Conference on Human-Computer Interaction (CHI'89), 1*, 241-245.

Holmback, H., Duncan, L., & Harrison, P. (2000). A word sense checking application for Simplified English. *Proceedings of the Third International Workshop on Controlled Language Applications,* Seattle. WA.

Holmback, H., Greaves, M., Bradshaw, J. (1999). A pragmatic principle for agent communication. In J. Bradshaw, O. Etzioni, & J. Mueller (Eds.), *Proceedings of Autonomous Agents '99*. Seattle, WA, New York: ACM Press.

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

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99),*159-166. New York: ACM Press.

Huang, X., Acero, A., Alleva, F., Hwang, M.-Y., Jiang, L., & Mahajan, M. (1995). Microsoft Windows highly intelligent speech recognizer: Whisper. *Proceedings of 1995 International Conference on Acoustics, Speech, and Signal Processing, 1,* 93-96.

Iverson, P., Bernstein, L., & Auer, E. (1998). Modeling the interaction of phonemic intelligibility and lexical structure in audiovisual word recogniton. *Speech Communication, 26(1-2),* 45-63. North Holland.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J., & Hinton, G.E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1), 79-87.

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 (COLING-ACL'98),* 624-630. University of Montreal Press.

Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A., & Smith, I. (1997). Unification-based multimodal integration. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 281-288. San Francisco, CA: Morgan Kaufmann.

Juang, B.-H., & Katagiri, S., (1992). Discriminative learning for minimum error classification, *IEEE Transactions on Signal Processing*, Vol. 40, No. 12, 3043-3054.

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 (CHI'99),* 568-575. ACM Press.

Karshmer, A. I. & Blattner, M. (organizers). (1998). *Proceedings of the Third International ACM Proceedings of the Conference on Assistive Technologies (ASSETS'98).* Marina del Rey, CA. (URL http://www.acm.org/sigcaph/assets/assets98/assets98index.html).

Katagiri, S., & McDermott, E. (in press). Discriminative training — recent progress in speech recognition. *Handbook of Pattern Recognition and Computer Vision (2nd Edition).* (Eds. C.H. Chen, L.F. Pau, & P.S.P. Wang). World Scientific Publishing Company.

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

Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Office Information Systems, 2(1)*, 26-41.

Kempf, J. (1994). *Preliminary handwriting recognition engine application program interface for Solaris 2* . Mountain View, CA: Sun Microsystems.

Kendon, A. (1980). Gesticulation and speech: Two aspects of the process of utterance. In M. Key (Ed.), *The Relationship of Verbal and Nonverbal Communication* (pp 207-227). The Hague: Mouton.

Klemmer, S., Sinha, A., Chen, J, Landay, J., Aboobaker, A. (2000). SUEDE: A wizard of oz prototyping tool for speech user interfaces. To appear in *Proceedings of User Interface Software and Technology 2000*, San Diego, California, November 2000.

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

Lai, J., & Vergo, J. (1997). MedSpeak: Report creation with continuous speech recognition. *Proceedings of the Conference on Human Factors in Computing (CHI'97),* 431- 438. ACM Press.

Landay, J. A. (1996). *Interactive sketching for the early stages of user interface design.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA.

Landay, J. A., & Myers, B. A. (1995). Interactive sketching for the early stages of user interface design. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'95)*, 43-50. New York: AMC Press.

Landay, J. A., & Myers, B. A. (1996). Sketching storyboards to illustrate interface behavior. *Human Factors in Computing Systems (CHI'96), Conference Companion*, 193-194. New York: AMC Press.

Larson, J. A., Oviatt, S. L., & Ferro, D. (1999). Designing the user interface for pen and speech applications. *CHI '99 Workshop, Conference on Human Factors in Computing Systems (CHI'99)*, Philadelphia, Pa.

Lester, J., Voerman, J., Towns, S., & Callaway, C. (1999). Deictic believability: Coordinated gesture, locomotion, and speech in lifelike pedagogical agents. *Applied Artificial Intelligence*, *13*, 383-414.

Levelt, W., Richardson, G., & Heu, W. (1985). Pointing and voicing in deictic expressions. *Journal of Memory and Language*, *24*, 133-164.

Markinson, R. (1993). Personal communication, University of California at San Francisco Medical School.

Martin, A., Fiscus, J., Fisher, B., Pallett, D. & Przybocki, M. (1997). System descriptions and performance summary. *Proceedings of the Conversational Speech Recognition Workshop/DARPA Hub-5E Evaluation.* Morgan Kaufman.

Martin, D. L., Cheyer, A. J., & Moran, D. B. (1999). The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, *13,* 91-128.

McGee, D., Cohen, P. R., & Oviatt, S. L. (1998). Confirmation in multimodal systems. *Proceedings of the International Joint Conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics* (COLING-ACL '98), 823-829. University of Montreal Press.

McGee, D.R., Cohen, P.R., & Wu, L., (2000). Something from nothing: Augmenting a paper-based work practice with multimodal interaction. *Proceedings of the Designing Augmented Reality Environments Conference 2000*, 71-80. Copenhagen, Denmark: ACM Press.

McGurk, H., & MacDonald, J. (1976). Hearing lips and seeing voices. *Nature, 264*, 746-748.

McNeill, D. (1992). *Hand and mind: What gestures reveal about thought.* Chicago, IL: University of Chicago Press.

Meier, U., Hurst, W., & Duchnowski, P. (1996). Adaptive bimodal sensor fusion for automatic speechreading. *Proceedings of International Conference on Acoustic, Speech and Signal Processing 2*, 833-836. IEEE Press.

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.

Nicolino, T. (1994). A natural language processing based situation display. *Proceedings of th*e *1994 Symposium on Command and Control Research and Decision Aids*, 575-580. Monterey, CA: Naval Postgraduate School.

Oviatt, S. L. (1992). Pen/voice: Complementary multimodal communication. *Proceedings of Speech Tech'92*. New York, NY.

Oviatt, S. L. (1996). User-centered design of spoken language and multimodal interfaces. *IEEE Multimedia* (special issue on *Multimodal Interaction*), *3(4)*, 26-35. (Reprinted in M. Maybury & W. Wahlster (Eds.), *Readings on Intelligent User Interfaces*. Morgan-Kaufmann.)

Oviatt, S. L. (1997). Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction* (special issue on *Multimodal Interfaces*), *12*, 93-129.

Oviatt, S. L. (1999a). Mutual disambiguation of recognition errors in a multimodal architecture. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99)*, 576-583. New York: ACM Press.

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

Oviatt, S.L. (in press). Harnessing new media in multimodal systems, *Communications of the ACM*, September 2000.

Oviatt, S.L., Cohen, P.R. (2000). Multimodal systems that process what comes naturally. *Communications of the ACM*, 43(3), 45-53.

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. (1991). Discourse structure and performance efficiency in interactive and noninteractive spoken modalities. *Computer Speech and Language, 5(4),* 297-326.

Oviatt, S. L., Cohen, P. R., Fong, M. W., & Frank, M. P. (1992). A rapid semi-automatic simulation technique for investigating interactive speech and handwriting. *Proceedings of the International Conference on Spoken Language Processing*, *2*, 1351-1354. University of Alberta.

Oviatt, S. L., Cohen, P. R., & Wang, M. Q. (1994). Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity. *Speech Communication*, *15,* 283-300. European Speech Communication Association.

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)*, 415-422. New York: ACM Press.

Oviatt, S. L., & Kuhn, K. (1998). Referential features and linguistic indirection in multimodal language. *Proceedings of the International Conference on Spoken Language Processing, 6*, 2339-2342. Syndey: ASSTA, Inc.

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, (pp. 551-554). Acoustical Society of Japan.

Oviatt, S. L., & Pothering, J. (1998). Interacting with animated characters: Research infrastructure and next-generation interface design. *Proceedings of the First Workshop on Embodied Conversational Characters*, 159-165.

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. University of Delaware Press.

Pankanti, S., Bolle, R.M., & Jain, A. (Eds.) (2000). Biometrics: The future of identification (special issue), *Computer*, 2000, 33(2), 46-80.

Papineni, K. A., Roukos, S., & Ward, R. T. (1997). Feature-based language understanding. *Proceedings of the 5th European Conference On Speech Communication and Technology, 3*, 1435-1438. Rhodes, Greece: European Speech Communication Association.

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. Menlo Park, CA: AAAI Press.

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*, 121-124. IEEE Press.

Pentland, A. (1996). Smart rooms. *Scientific American,* April, pp. 68-76.

Perrault, C. R. (1990). An application of default logic to speech act theory. In P. R. Cohen, J. Morgan, & M. E. Pollack, (Eds.), *Intentions in communication* (pp. 161-186). Cambridge, MA: MIT Press.

Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE Proceedings*, 1989, 267-296.

Reithinger, N., & Klesen, M. (1997). Dialogue act classification using language models. *Proceedings of Eurospeech'97*, 2235-2238.

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

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

Rogozan, A., & Delegise, P. (1998). Adaptive fusion of acoustic and visual sources for automatic speech recognition. *Speech Communication*, 26 (1-2), 149-161.

Roth, S. F., Mattis, J., & Mesnard, X. (1991). Graphics and natural language as components of automatic explanation. In J. W. Sullivan & S. W. Tyler (Eds.), *Intelligent User Interfaces* (pp. 207-239). New York: ACM Press, Frontier Series.

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

Rudnicky, A., & Hauptman, A. (1992). Multimodal interactions in speech systems. In M. Blattner & R. Dannenberg (Eds.), *Multimedia Interface Design* (pp.147-172). New York, NY: ACM Press, Frontier Series.

Sadek, D. (1991). Dialogue acts are rational plans. *Proceedings of the ESCA/ETRW Workshop on the Structure of Multimodal Dialogue.* Maratea, Italy.

Sag, I.A., & Wasow, T. (1999). *Syntactic theory: A formal introduction*. Stanford: CSLI Publications.

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

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language.* Cambridge: Cambridge University Press.

Sejnowski, T., Yuhas, B., Goldstein, M., & Jenkins, R.(1990). Combining visual and acoustic speech signal with a neural network improves intelligibility. In D. Touretzky (Ed.), *Advances in Neural Information Processing Systems,* (pp. 232-239).

Seneff, S., Hurley, E., Lau, R., Pao, C., Schmid, P., & Zue, V. (1998). Galaxy-II: a reference architecture for conversational system development, *Proceedings of the International Conference on Spoken Language Processing*. Sydney, Australia.

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 (PUI'97),* 91-93. Banff, Canada.

Spiegel, M., & Kamm, C. (Eds.), (1997). Special issue on Interactive Voice Technology for Telecommunications Applications (IVTTA'96), *Speech Communication,* 23(1-2). North Holland.

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. Germany: Shaker Verlag.

Suhm, B., Myers, B., & Waibel, A. (1996). Interactive recovery from speech recognition errors in speech user interfaces. *Proceedings of the International Conference on Spoken Language Processing,* 2, 861-864. University of Delaware Press.

Sun. (1998). *Java Speech API.* Palo Alto, CA: Sun Microsystems.

Sutton, S., & Cole, R. (1997). The CSLU Toolkit: Rapid prototyping of spoken language systems. *Proceedings of UIST '97: the ACM Symposium on User Interface Software and Technology*, 85-86. New York: ACM Press.

Turk, M., & Robertson, G. (Eds.) (2000). Perceptual user interfaces (special issue), *Communications of the ACM*, 2000, 43(3), 32-70.

Vergo, J. (1998). A statistical approach to multimodal natural language interaction. *Proceedings of the AAAI'98 Workshop on Representations for Multimodal Human-Computer Interaction*, 81-85. AAAI Press.

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. IEEE Press.

Vo, M. T. (1998). *A framework and toolkit for the construction of multimodal learning interfaces.* Unpublished Ph.D. thesis, Carnegie Mellon University.

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*.

(URL http://werner.ira.uka.de/ISL.publications.html#1995)

Wagner, A. (1990). Prototyping: A day in the life of an interface designer. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design* (pp. 79-84). Reading, MA: Addison-Wesley.

Wahlster, W., Andre, E., W., Finkler, W., Profitlich, H.-J., & Rist, T. (1993). Plan-based integration of natural language and graphics generation. *Artificial Intelligence*, *63*, 387-427.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K.J., (1989). Phenome recognition using time-delay neural networks. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, 328-339.

Waibel, A., Vo, M. T., Duchnowski, P., & Manke, S. (1995). Multimodal interfaces. (Special Volume on Integration of Natural Language and Vision Processing) *Artificial Intelligence Review, 10 (3-4)*, 299-319. Kluwer Academic Publishers.

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*. 3938-3942. IEEE Press.

Wickens, C. D., Sandry, D. L., & Vidulich, M. (1983). Compatibility and resource competition between modalities of input, central processing, and output. *Human Factors*, 25, 227-248.

Wickens, C. D., Vidulich, M., & Sandry-Garza, D. (1984). Principles of S-R-C compatibility with spatial and verbal tasks: The role of display-control interfacing. *Human Factors*, 26, 533-534.

Wojcik, R., & Holmback, H. (1996). Getting a controlled language off the ground at Boeing. *Proceedings of the First International Workshop on Controlled Language Applications*, 22-31.

Wolf, C. G., & Morrel-Samuels, P. (1987). The use of hand-drawn gestures for text editing. *International Journal of Man-Machine studies, 27*, 91-102. Academic Press.

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

Wu, L., Oviatt, S., & Cohen, P. (in submission). From members to teams to committee: A robust approach to gestural and multimodal recognition.

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99).* ACM Press: New York, 246-253.