Declaration of Rachel J. Watters on Authentication of Publication

I, Rachel J. Watters, am a librarian, and the Head of Resource Sharing for the General Library System, Memorial Library, located at 728 State Street, Madison, Wisconsin, 53706. Part of my job responsibilities include oversight of Wisconsin TechSearch ("WTS"), an interlibrary loan department at the University of Wisconsin-Madison. I have worked as a librarian at the University of Wisconsin library system since 1998, starting as a graduate student employee in the Kurt F. Wendt Engineering Library and WTS, then as a librarian in Interlibrary Loan at Memorial Library. I began professional employment at WTS in 2002 and became WTS Director in 2011. In 2019, I became Head of Resource Sharing for UW-Madison's General Library System. I have a master's degree in Library and Information Studies from the University of Wisconsin-Madison. Through the course of my studies and employment, I have become well informed about the operations of the University of Wisconsin library system, which follows standard library practices.

This Declaration relates to the dates of receipt and availability of the following:

> **Oviatt, S., Cohen, P., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J., and Ferro, D. (2000). Designing the user interface for multimodal speech and pen-based gesture applications: state-of-the-art systems and future research directions. *Human-Computer Interaction, 15*(4), 263-322.**

Standard operating procedures for materials at the University of Wisconsin-Madison Libraries. When an issue was received by the Library, it would be checked in,

Amazon/VB Assets
Exhibit 1006
Page 1

stamped with the date of receipt, added to library holdings records, and made available to readers as soon after its arrival as possible. The procedure normally took a few days or at most 2 to 3 weeks.

Exhibit A to this Declaration is a true and accurate copy of the front matter and page 263 with library date stamp of *Human-Computer Interaction* (2000), from the University of Wisconsin-Madison Library collection. Exhibit A also includes an excerpt of pages 263 to 322 of that issue, showing the article entitled *Designing the user interface for multimodal speech and pen-based gesture applications: state-of-the-art systems and future research directions* (2000). Based on this information, the date stamp on page 263 indicates *Designing the user interface for multimodal speech and pen-based gesture applications: state-of-the-art systems and future research directions* (2000) was received by the Kurt F. Wendt Library, University of Wisconsin-Madison on April 25, 2001.

Based on the information in Exhibit A, it is clear that the issue was received by the library on or before April 25, 2001, catalogued and available to library patrons within a few days or at most 2 to 3 weeks after April 25, 2001.

Members of the interested public could locate the *Human-Computer Interaction* (2000) publication after it was cataloged by searching the public library catalog or requesting a search through WTS. The search could be done by title and/or subject key

words. Members of the interested public could access the publication by locating it on the library's shelves or requesting it from WTS.

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Date: March 13, 2025

Rachel J. Watters
Head of Resource Sharing

Memorial Library
728 State Street
Madison, Wisconsin 53706

HUMAN–COMPUTER INTERACTION

A Journal of Theoretical, Empirical, and Methodological Issues of User Science and of System Design

S
H 88
C 737

Volume 15, Number 4
2000

LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
Mahwah, New Jersey London

Amazon/VB Assets
Exhibit 1006
Page 4

HUMAN–COMPUTER INTERACTION

EDITOR: Thomas P. Moran, *Xerox Palo Alto Research Center*

ADMINISTRATIVE EDITOR: Patricia Sheehan, *Xerox Palo Alto Research Center*
PRODUCTION SUPERVISOR: Amy E. Munno, *Lawrence Erlbaum Associates, Inc.*

EDITORIAL BOARD:

John Anderson ..... *Carnegie Mellon University, Pittsburgh, PA*
Ruven Brooks..... *Rockwell Software, West Allis, WI*
John Seely Brown ..... *Xerox Palo Alto Research Center, Palo Alto, CA*
John M. Carroll ..... *Virginia Polytechnic Institute & State University, Blacksburg, VA*
Bill Curtis ..... *TeraQuest Metrics, Inc., Austin, TX*
Susan T. Dumais..... *Microsoft Research, Redmond, WA*
Jonathan Grudin..... *Microsoft Research, Redmond, WA*
Robert E. Kraut ..... *Carnegie Mellon University, Pittsburgh, PA*
Thomas Landauer ..... *University of Colorado, Boulder, CO*
Clayton Lewis..... *University of Colorado, Boulder, CO*
Scott MacKenzie ..... *York University, Toronto, Canada*
Brad A. Myers ..... *Carnegie Mellon University, Pittsburgh, PA*
Donald A. Norman ..... *Nielsen Norman Group, Atherton, CA*
Dan R. Olsen, Jr. ..... *Brigham Young University, Provo, UT*
Gary M. Olson ..... *University of Michigan, Ann Arbor, MI*
Judith S. Olson ..... *University of Michigan, Ann Arbor, MI*
Sharon Oviatt ..... *Oregon Graduate Institute of Science & Technology, Portland, OR*
Stephen J. Payne ..... *University of Wales, Cardiff, UK*
Richard Pew..... *Bolt, Beranek, & Newman, Inc., Cambridge, MA*
James R. Rhyne..... *IBM T. J. Watson Research Center, Yorktown Heights, NY*
Elliot Soloway..... *University of Michigan, Ann Arbor, MI*
Lucy Suchman..... *Xerox Palo Alto Research Center, Palo Alto, CA*
Steve Whittaker ..... *AT&T Labs–Research, Florham Park, NJ*
Terry Winograd ..... *Stanford University, Stanford, CA*
Richard Young..... *University of Hertfordshire, Hatfield, UK*

Subscription Information. *Human–Computer Interaction* is published quarterly by Lawrence Erlbaum Associates, Inc. Subscriptions are available only on a calendar-year basis. **Printed** journal subscription rates are $50 (individuals) and $320 (institutions) within the United States and Canada; $80 (individuals) and $350 (institutions) outside the United States and Canada. Order printed subscriptions through the Journal Subscription Department at Lawrence Erlbaum Associates (address at bottom). Claims for missing issues cannot be honored beyond 4 months after mailing date. Duplicate copies cannot be sent to replace issues not delivered due to failure to notify publisher of change of address.

Electronic. Full-price print subscribers to Volume 15, 2000 are entitled to receive the electronic version free of charge. *Electronic-only* subscriptions are available at a reduced subscription price. Please visit the LEA Web site at http://www.erlbaum.com for complete information.

This journal is abstracted or indexed in: *Cambridge Scientific Abstracts: Health & Safety Science Abstracts, Risk Abstracts; CompuMath Citation Index; Computer Abstracts; Educational Technology Abstracts; Engineering Information, Inc.; Ergonomics Abstracts; ERIC Clearinghouse in Education; ISI: SciSearch, Research Alert, Current Contents/Engineering, Computing and Technology; Knowledge Engineering Review; Library and Information Science Abstracts; PsycINFO/Psychological Abstracts; Sociological Abstracts.* Microform copies of this journal are available through Bell & Howell Information and Learning, P.O. Box 1346, Ann Arbor, MI 48106–1346.

Special requests for permission should be addressed to the Permissions Department at Lawrence Erlbaum Associates (address at bottom).

Copyright © 2000, Lawrence Erlbaum Associates, Inc. No part of this publication may be used, in any form or by any means, without permission from the publisher.

Printed in the United States
ISSN 0737–0024

Lawrence Erlbaum Associates, Inc., 10 Industrial Avenue, Mahwah, NJ 07430–2262

Amazon/VB Assets
Exhibit 1006
Page 5

HUMAN-COMPUTER INTERACTION, 2000, Volume 15, pp. 263–322
Copyright © 2000, Lawrence Erlbaum Associates, Inc.
COLLEGE OF ENGINEERING
APR 23 2001
UW-MADISON, WI 5370

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

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

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 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, multiagent 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 human–computer interaction, 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

Amazon/VB Assets
Exhibit 1006
Page 6

CONTENTS

1. INTRODUCTION TO MULTIMODAL SPEECH AND PEN-BASED GESTURE INTERFACES
2. ADVANTAGES AND OPTIMAL USES OF MULTIMODAL INTERFACE DESIGN
3. ARCHITECTURAL APPROACHES TO MULTIMODAL PEN–VOICE INTEGRATION AND SYSTEMS
   3.1. Introduction to Multimodal Architectural Requirements
   3.2. Multiagent Architectures and Multimodal Processing Flow
   3.3. Frame-Based and Unification-Based Multimodal Integration
   3.4. New Hybrid Architectures: An Illustration
4. DIVERSITY OF EMERGING SPEECH AND GESTURE APPLICATIONS
   4.1. OGI's QuickSet Multimodal System
   4.2. IBM's Human–Centric Word Processor
   4.3. Boeing's Virtual Reality Aircraft Maintenance Training Prototype
   4.4. NCR's Field Medic Information System
   4.5. BBN's Portable Voice Assistant
   4.6. Limitations of Current Speech and Gesture Multimodal Systems
5. FUTURE RESEARCH DIRECTIONS FOR MULTIMODAL INTERFACES
   5.1. Cognitive Theory and Empirical Science Underpinnings
   5.2. New Multimodal Interface Concepts
   5.3. Multimodal Language and Dialogue Processing
   5.4. Error-Handling Techniques
   5.5. Adaptive Multimodal Architectures
   5.6. Multidevice, Multiuser Ubiquitous Computing
   5.7. Multimodal Research Infrastructure
      Tools for Multimodal System Design, Iteration, and Corpus Collection
      Tools for Multimodal Software Development
6. CONCLUSIONS

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.

Amazon/VB Assets
Exhibit 1006
Page 7

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, be usable by a broader spectrum of the average population, and function more reliably under realistic and challenging usage conditions. In this article, 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 multiperson use. Before this new class of systems can proliferate, toolkits also will be needed to promote software development for both simulated and functioning systems.

1. INTRODUCTION TO MULTIMODAL SPEECH AND PEN-BASED 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 they are preferred by users for many applications. They have the potential to expand computing to more challenging applications, be used by a broader spectrum of everyday people, and 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.

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

Amazon/VB Assets
Exhibit 1006
Page 8

multimodal systems that process complex gestural input other than just pointing, examples of which are outlined in Section 4 of this article. Since Bolt's early inspiration, the basic architectural components and framework needed to support more general multimodal systems have become better established, as is 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 (SLT) 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 Defense Advanced Research Projects Agency's (DARPA's) Speech Grand Challenge project and similar international efforts (Cole et al., 1997; A. Martin, Fiscus, Fisher, Pallett, & Przybocki, 1997), there has been significant progress in the development of SLT. Spoken-language systems now are implemented extensively for telephony applications (Spiegel & Kamm, 1997) and on workstations, and they are beginning to appear on palm-held computers. These new technical capabilities, along with advances in natural language processing (NLP), 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 SLT, 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 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 article, we introduce multimodal speech and pen-based gesture interfaces, with a focus in Section 2 on their primary advantages and optimal

Amazon/VB Assets
Exhibit 1006
Page 9

uses. To date, multimodal systems that combine either speech and pen input (Oviatt & Cohen, 2000) or speech and lip movements (Benoit, Martin, Pelachaud, Schomaker, & Suhm, 2000; Rubin, Vatikiotis-Bateson, & Benoit, 1998; Stork & Hennecke, 1995) constitute the two major research areas within the field. Although many of the issues discussed for multimodal systems incorporating speech and two-dimensional pen gestures also are relevant to those involving continuous three-dimensional (3D) 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 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 illustrate five diverse state-of-the-art multimodal systems that support challenging applications. These include map-based and virtual reality (VR) 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 multiperson 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, Oviatt, & Ferro, 1999). A multimodal interface offers the user freedom to use a combination of modalities or switch to a better-suited modality, depending on the specifics of the task or environment. Because 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

Amazon/VB Assets
Exhibit 1006
Page 10

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, and 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, 1997; Oviatt & Olsen, 1994; Suhm, 1998).

Although supporting portability, pen input also has a somewhat different and multifunctional range of capabilities. Although the pen can be used to write words that are analogous to speech, it also can be used to render signatures and to convey symbols and signs (e.g., digits, abbreviations), gestures, simple graphics, and artwork. 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 (Gong, 1995; Holzman, 1999). 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 easily can 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 shorter and more simplified linguistic constructions with fewer locative descriptions (Oviatt, 1997; Oviatt & Kuhn, 1998). This constellation of multimodal performance advantages cor-

Amazon/VB Assets
Exhibit 1006
Page 11

responds with a 90 to 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 because 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 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, in terms of both error avoidance and graceful recovery from errors (Oviatt, 1999a; Oviatt, Bernard, & Levow, 1999; Oviatt & vanGent, 1996; 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 NLP 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 et al., 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.

Because there are large individual differences in ability and preference to use different modes of communication, a multimodal interface permits the

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.

Amazon/VB Assets
Exhibit 1006
Page 12

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 and text-to-speech output, as may a manually impaired user (e.g., with repetitive stress injury or an arm in a cast). In contrast, a user with a hearing impairment, strong accent, or a cold might 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 (R. Markinson, personal communication, 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 might 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 do the following:

- 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 NLP.
- 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.

Amazon/VB Assets
Exhibit 1006
Page 13

- 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 PEN–VOICE 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 (1980) Put That There system, 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. Although such an approach suffices for processing a *point-and-speak multimodal integration pattern*, unfortunately, fewer than 20% of all users' multimodal commands are of this limited type (McNeill, 1992; Oviatt et al., 1997). 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., a 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 a variety of multimodal speech-and-gesture integration patterns and 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.

Amazon/VB Assets
Exhibit 1006
Page 14

Multimodal systems designed to handle joint processing of input signals have 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 (HMMs) or temporal neural networks (NNs).[2] Examples of representative research include Bregler, Manke, Hild, and Waibel (1993); Pavlovic, Berry, and Huang (1997); Pavlovic and Huang (1998); and Vo et al. (1995). 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 (Rubin et al., 1998; Stork & Hennecke, 1995), for which both input channels provide corresponding information about the same articulated phonemes and words. However, such a system tends to not 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 typically is 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 retraining. 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., 1989/1998), QuickSet (Cohen, Johnston, et al., 1997), CUBRICON (Neal & Shapiro, 1991), Virtual World

2. State-of-the-art HMMs have been applied widely to problems such as large-vocabulary continuous-speech recognition and handwriting recognition (Rabiner, 1989). NNs are an alternative pattern recognition technique, and temporal NNs are ones capable of modeling the temporal structure of input signals, such as time-delay NNs (Waibel, Hanazawa, Hinton, Shikano, & Lang, 1989) and recurrent NNs (Elman, 1990).

Amazon/VB Assets
Exhibit 1006
Page 15

(Codella et al., 1992), Finger-Pointer (Fukumoto, Suenaga, & 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. Because 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 et al., 1997). Empirical work on speech and gesture input has established that users' written input precedes speech during a sequentially integrated multimodal command (Oviatt et al., 1997), and it also has clarified the distribution of typical intermodal 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 intermodal 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 on and used to overcome weaknesses in the other (Cohen et al., 1989/1998; Oviatt, 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, 2000b).

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 the following:

- 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.

Amazon/VB Assets
Exhibit 1006
Page 16

- 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. Multiagent Architectures and Multimodal Processing Flow

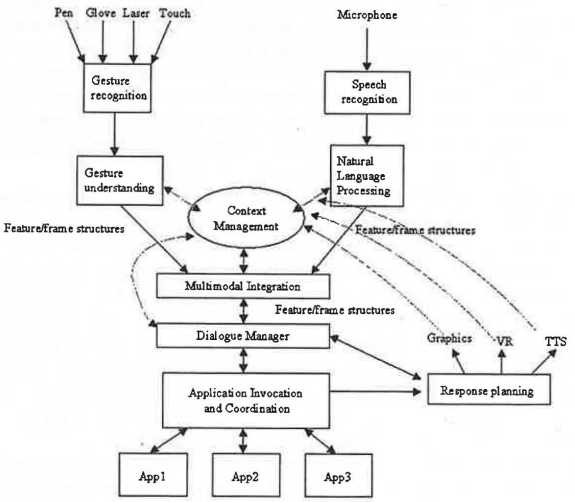
Before discussing the motivation and specifics of multimodal speech and pen-based gesture architectures, it is important to identify the primary ways in which emerging multimodal architectures are distinct from those of standard GUIs. First, GUIs typically assume that there is a single event stream that 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 Personal Digital Assistants may extract features from speech input and transmit them to a recognizer that resides on a server, as does Bolt, Beranek, & Newman's (BBN's) Portable Voice Assistant (PVA; 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

3. Because this article 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.

Amazon/VB Assets
Exhibit 1006
Page 17

Figure 1. **Typical information flow in a multimodal architecture.**



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

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 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.

There are numerous ways to realize this information-processing flow as an architecture. One well-understood way is to pipeline the components via proce-

Amazon/VB Assets
Exhibit 1006
Page 18

dure 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), this method might 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 *multiagent architectures*, such as the Open Agent Architecture[4] (OAA; Cohen, Cheyer, Wang, & Baeg, 1994/1997; D. L. Martin, Cheyer, & Moran, 1999) and Adaptive Agent Architecture (Kumar & Cohen, 2000).

In a multiagent 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, multicasting, and other concepts from distributed systems. In some multiagent architectures, agents communicate directly with other components about which they have knowledge (e.g., component's name, Application Program Interface). This design has the advantage of no intermediaries, but it can be brittle in the face of agent failure.

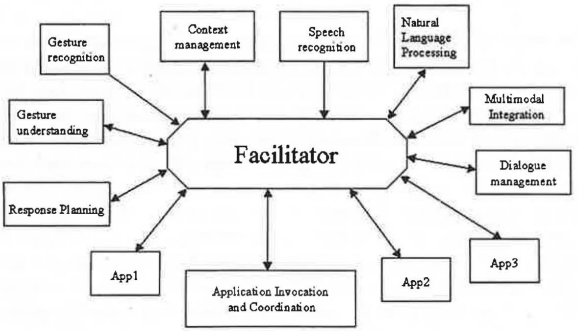
As an alternative design, many architectures have adopted a *facilitated* form of communication in which agents do not need to know to whom they are making requests or supplying answers. Instead, these agents communicate through a known facilitator, which routes messages to the interested and capable 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 because the facilitator can route communications to multiple agents that might be interested in the same messages.

Figure 2 illustrates the same basic components as Figure 1 but 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 or hub can be a bottleneck, possibly impeding high-volume multimedia data transfer (e.g., speech). It also is a single point of failure that 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.

4. OAA is a trademark of SRI International.

Amazon/VB Assets
Exhibit 1006
Page 19

Figure 2. **Facilitated multimodal architecture.**



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 might have arrived from the same user or other users is processed asynchronously. Processing within this kind of a multiagent framework is usually bottom-up and asynchronous, with an emergent control flow.[5]

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 Communicator's architecture, the hub also is a single point of failure.

Amazon/VB Assets
Exhibit 1006
Page 20

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

To fuse information derived from multiple modalities, various research groups (Cheyer & Julia, 1995; Pavlovic & Huang, 1998; Shaikh et al., 1997; Vo & Wood, 1996) independently have 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 FSs have been developed (Calder, 1987; Carpenter, 1990, 1992). These structures are pervasive in NLP and have been used for lexical entries, grammar rules, and meaning representations (Sag & Wasow, 1999). For FS logics, the primary operation is unification–a more general approach that subsumes other frame-merging strategies. Unification-based architectures only recently have been applied to multimodal system design (Cohen, Johnston, et al., 1997; Johnston et al., 1997; Wu, Oviatt, & Cohen, 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. FS unification is a generalization of term unification in logic programming languages, such as Prolog. An 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 FS. When two FSs, 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 FS. If both are variables, they become bound together, which constrains them to always receiving the same value.

Amazon/VB Assets
Exhibit 1006
Page 21

FSs 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 FSs, the unification operation is applied recursively.[6] Importantly, FS unification can result in a directed acyclic graph structure when more than one value in the collection of feature–value pairs makes use of the same variable. Whatever value ultimately is 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.

FS unification is suited ideally 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 FSs to be unified and on the development of canonical meaning representations that are common across input modes and research sites.

3.4. New Hybrid Architectures: An Illustration

When statistical processing techniques are combined with a symbolic unification-based approach that merges FSs, 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 (MTC) techniques (Wu, Oviatt, & Cohen, 1998; Wu et al., 1999). QuickSet is 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 (a) recognition accuracy of the individual modes, (b) the mapping structure between multimodal commands and their speech–pen constituents, (c) the man-

6. FS unification differs from term unification in logic programming in that the features are encoded positionally in a term, whereas they are explicitly labeled and unordered in an FS. Otherwise, the unification of values is identical.

7. Multimodal architectures also can be hybrids in the sense of combining HMMs and NNs. This can be an opportune combination in the case of a pen–voice system, because speech is processed well with HMMs and NNs handle pen input well.

Amazon/VB Assets
Exhibit 1006
Page 22

ner of combining posterior probabilities, and (d) the prior distribution of multimodal commands. The Associative Mapping and 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, because a constituent in one 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 FSs that cannot possibly be unified semantically. As such, it provides an efficient means of quickly ruling out impossible unifications.

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, who then applies various weighting parameters to their reported scores. Furthermore, each team can apply a different weighting scheme and 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.

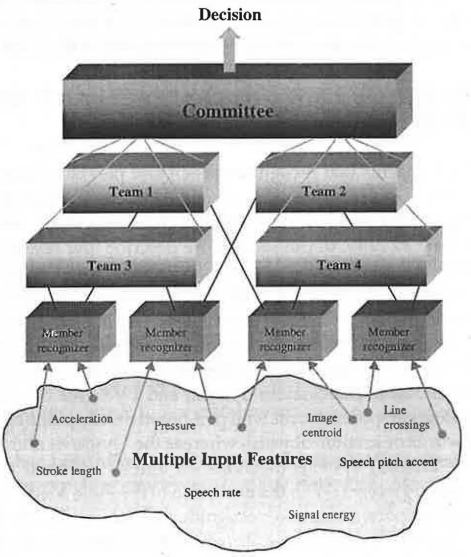
MTC serves two purposes within the QuickSet system. It is used as a gesture recognizer, and in this case the members are recognizers for the 190 possible gestures. It also is used as the statistical mechanism for combining recognition scores from the unifiable speech and gesture FSs.

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

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

Amazon/VB Assets
Exhibit 1006
Page 23

Figure 3. **The three-tiered bottom-up Members–Teams–Committee architecture, composed of multiple members, multiple teams, and a decision-making committee.**



the merged FSs is the cross-product of the probabilities of individual FSs. In the approach presented here, the probability of the merged FS is the weighting interpolation of the probabilities of individual FSs. 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 et al., 1999).[9] The favorable performance of the MTC architectural approach can be attributed to a variety of fac-

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) described a method for estimating a multimodal system's upper and lower performance bounds.

Amazon/VB Assets
Exhibit 1006
Page 24

tors, 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), and normalization of recognizer output from speech and gesture to the same scale (see details in Wu et al., 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 (HCWP), Field Medic Information System, and PVA (see Sections 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 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 are described in greater detail in this section, as well as their main functionality and architectural features.

4.1. OGI's QuickSet Multimodal 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,

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).

Amazon/VB Assets
Exhibit 1006
Page 25

and direct manipulation (Cohen, Johnston, 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 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.

When used together, speech and gesture input are interpreted by parallel recognizers, which generate a set of typed-FS 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 et al., 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 OAA as its infrastructure (Cohen et al., 1994/1997; D. L. Martin et al., 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, and so forth to agents that have advertised relevant capabilities. The facilitator also supports triggering, thereby enabling asynchronous communication.

Among the agents that comprise QuickSet's main components[11] are (a) 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; (b) the MTC gesture recognizer (Wu et al., 1999); (c) a multimodal natural language parser (Johnston, 1998); (d) a unification-based multimodal integration component (Johnston et al., 1997); and (e) 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

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.

Amazon/VB Assets
Exhibit 1006
Page 26

Figure 4. **Functionality, architectural features, and general classification of different multimodal speech and pen-based gesture systems.**

Multimodal System Characteristics	QuickSet	Human-Centric Word Processor	VR Aircraft Maintenance Training	Field Medic Information	Portable Voice Assistant
Recognition of simultaneous or alternative individual modes	Simultaneous and individual modes	Simultaneous and individual modes	Simultaneous and individual modes	Alternative individual modes[a]	Simultaneous and individual modes
Type and size of gesture vocabulary	Pen input, multiple gestures, large vocabulary	Pen input, deictic selection	3D manual input, multiple gestures, small vocabulary	Pen input, deictic selection	Pen input, deictic selection[b]
Size of speech vocabulary[c] and type of linguistic processing	Moderate vocabulary, grammar based	Large vocabulary, statistical language processing	Small vocabulary, grammar based	Moderate vocabulary, grammar based	Small vocabulary, grammar based

Amazon/VB Assets
Exhibit 1006
Page 27

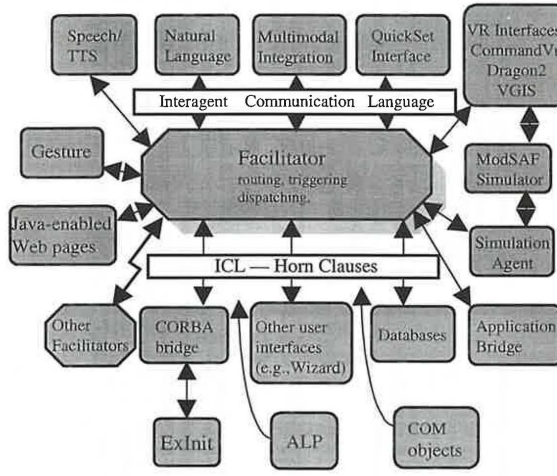
Type of signal fusion	Late semantic fusion, unification, hybrid symbolic/statistical MTC framework	Late semantic fusion, frame based	Late semantic fusion, frame based	No mode fusion	Late semantic fusion, frame based
Type of platform and applications	Wireless handheld, varied map and VR applications[d]	Desktop computer, word processing	Virtual reality system, aircraft maintenance training	Wireless handheld, medical field emergencies	Wireless handheld, catalogue ordering
Evaluation status	Proactive user-centered design and iterative system evaluations	Proactive user-centered design	Planned for future	Proactive user-centered design and iterative system evaluations	Planned for future

Note. VR = virtual reality; MTC = Members–Teams–Committee.
[a]The Field Medic Associate component recognizes speech only, and the Field Medic Coordinator (FMC) component recognizes gestural selections or speech. The FMC also can transmit digital speech and ink data, and can read data from smart cards and physiological monitors. [b]The Portable Voice Assistant also performs handwriting recognition. [c]A small speech vocabulary is up to 200 words, moderate is 201 to 1,000 words, and large is in excess of 1,000 words. For pen-based gestures, deictic selection is an individual gesture, a small vocabulary is 2 to 20 gestures, moderate is 21 to 100, and large is in excess of 100 gestures. [d]QuickSet's map applications include military simulation, medical informatics, real estate selection, and so forth.

Amazon/VB Assets
Exhibit 1006
Page 28

Figure 5. **QuickSet system running on a wireless handheld PC, showing user-created entities on the map interface.**

Figure 6. **The OGI QuickSet system's facilitated multiagent architecture.**



Amazon/VB Assets
Exhibit 1006
Page 29

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 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 & Oviatt, 1998). The agent architecture also enables the system to scale from hand-held to wall-size interfaces and 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 et al., 1997). During a recent case study involving a domain expert (i.e., U.S. Marine Corps major) who was engaged in an exercise initialization task, a ninefold reduction in entity creation time was demonstrated while he interacted multimodally using QuickSet, in comparison to 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, 3D virtual-terrain visualization systems, a community disaster management system, a medical informatics system, and so forth (Cohen et al., 1998; Cohen, Johnston, et al., 1997; Cohen et al., 1999; Oviatt, 1999a). QuickSet also has been transferred to the research laboratories of each of the U.S. armed services and many other government, commercial, and university sites, where it is being integrated with other software.

The componential nature of QuickSet's agent architecture has resulted in relatively easy reuse and integration of QuickSet's agents with other systems. For example, the QuickSet interface currently is being applied to a mobile augmented reality application (Feiner, MacIntyre, Hollerer, & Webster, 1997) and the Naval Research Laboratory's 3D Dragon 2 VR system (Cohen et al., 1999). In the latter interface, the user can speak while gesturing with a 6-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 and mobile environments (Oviatt, 2000a), and multimodal interfaces that support flexible interaction with tangible everyday objects (McGee & Cohen, 2001; McGee, Cohen, & Wu, 2000).

Amazon/VB Assets
Exhibit 1006
Page 30

4.2. IBM's Human-Centric Word Processor

The 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 (NLU) to control the application. In comparison, HCWP combines speech recognition and NLU (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 postdictation corrections is motivated by the facts that (a) people do not always dictate well-organized, grammatically correct text; (b) speech-recognition systems are imperfect, so not all words are transcribed correctly; and (c) 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 postdictation manipulation of text by permitting flexible multimodal interaction using spoken language, an electronic stylus, and NLP.[13]

Text editing is basically a spatial task that requires manipulation and movement of text elements. The vast majority of users, nearly 100%, prefer to interact multimodally when functioning in a visual–spatial domain. Some typical examples of multimodal constructions handled by HCWP include the following:

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 because executing this

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

Amazon/VB Assets
Exhibit 1006
Page 31

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 are 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 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 in either dictation or command mode. Mode switching usually is 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, Halverson, Horn, & Karat, 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.

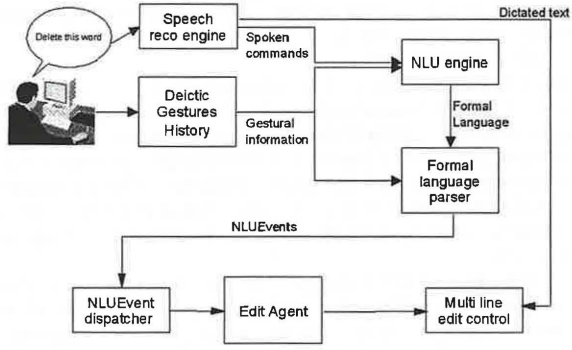
The HCWP system also uses an LCD tablet with a stylus for gestural input. A pen stylus is a natural, precise, and effective mode of interaction for spatially oriented tasks (Oviatt et al., 1997; Wolf & Morrel-Samuels, 1987). However, at present the only gesturing supported by the system is pointing. 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.

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.

Amazon/VB Assets
Exhibit 1006
Page 32

Figure 7. **Architectural flow of signal and language processing in IBM's Human-Centric Word Processor system.**



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.

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

- "Underline this sentence <pointing>. Make it red." (uses Principles 1, 5, & 3)
- <select some text> "Make it red." (uses Principle 2)
- "Move this sentence over here <points to sentence and new location>. Underline it." (uses Principles 1, 5, & 3)
- <select text ABC> "Underline it. Underline EFG." (uses Principles 2 & 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 subelements of formal language statements. Formal language statements produced by the NLU engine are flat-text repre-

Amazon/VB Assets
Exhibit 1006
Page 33

sentations, 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

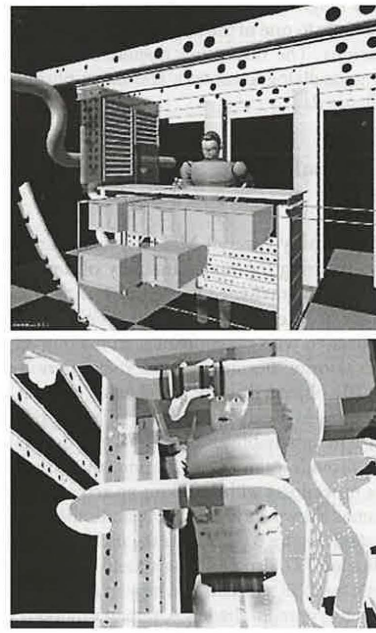
Boeing's Virtual Reality Aircraft Maintenance Training prototype is intended for use in assessing the maintainability of new aircraft designs and training mechanics in maintenance procedures using VR (Duncan, Brown, Esposito, Holmback, & Xue, 1999). A large percentage of the life cycle cost of an airplane is associated with maintenance, so making airplanes cheaper to maintain is a high priority. Because 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 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.

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

Amazon/VB Assets
Exhibit 1006
Page 34

Figure 8. **Boeing 777 aircraft's main equipment center.**



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 NLP 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. Because the Cyberglove is not

Amazon/VB Assets
Exhibit 1006
Page 35

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 (Holmback, 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.

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 VR system's "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 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.

The system configuration depicted here is realized as a distributed architecture using heterogeneous platforms (e.g., SGI Onyx, PCs, 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

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

Amazon/VB Assets
Exhibit 1006
Page 36

Figure 9. **Boeing's natural language understanding technology.**

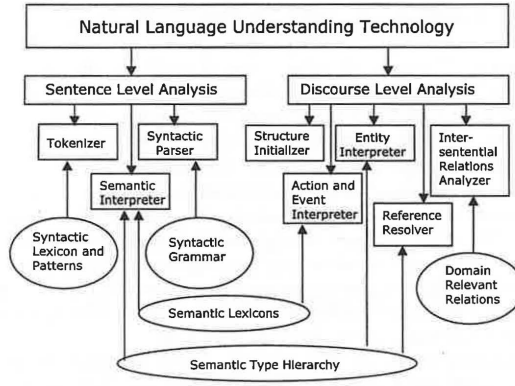
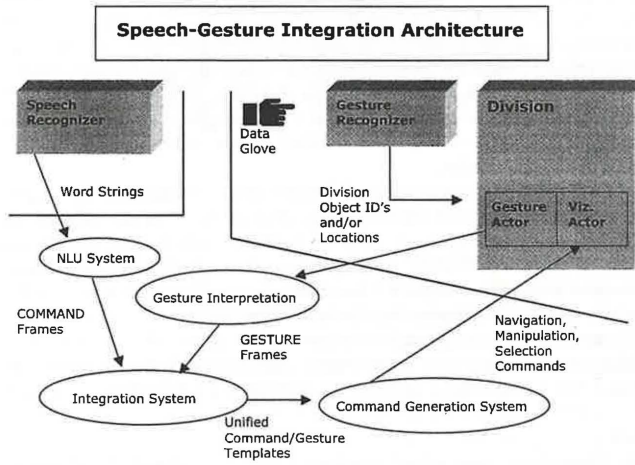


Figure 10. **Boeing's integrated speech and gesture system architecture.**



Amazon/VB Assets
Exhibit 1006
Page 37

these prototypes' limited domain functionality, add more discourse capabilities as well as natural language generation (NLG), 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 (e.g., see Holmback, Greaves, & Bradshaw, 1999). For this purpose, a full NLG system 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.

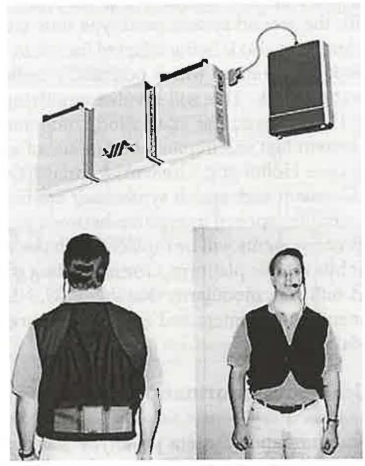
4.4. NCR's Field Medic Information System

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).

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 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

Amazon/VB Assets
Exhibit 1006
Page 38

Figure 11. **The NCR Field Medic Associate's (FMA) flexible hardware components (top); FMA inserted into medic's vest while he speaks during field use (bottom).**



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 known already. 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.

Amazon/VB Assets
Exhibit 1006
Page 39

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 permits neither 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 number), and 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.

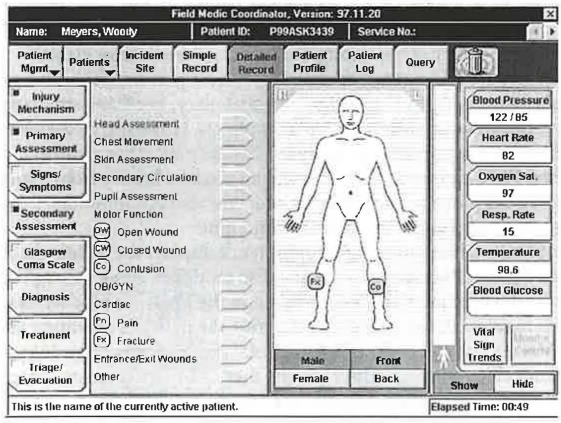
The FMC is a handheld tablet computer. It displays the patient record illustrated in Figure 12 and permits the medic to modify it, either through the same speech-audio user interface incorporated in the FMA, or by a series of quick pen taps to select items on lists that describe the patient's condition 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 cannot 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 relay those records to a hospital via cellular telephone, satellite, or radio while 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, and it does not 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.

Amazon/VB Assets
Exhibit 1006
Page 40

Figure 12. **The NCR Field Medic Coordinator's visual interface.**



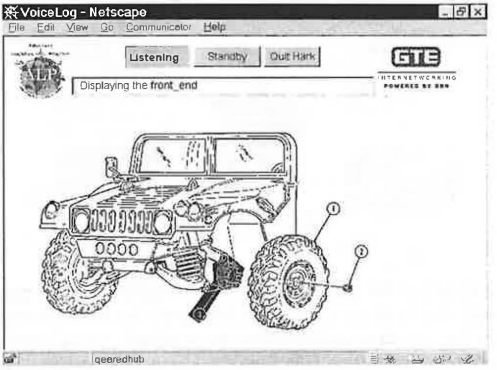
4.5. BBN's Portable Voice Assistant

The 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 online 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 are 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 software was used. This system handles selection gestures on the graphic images, as well as

Amazon/VB Assets
Exhibit 1006
Page 41

Figure 13. **The BBN Portable Voice Assistant interface illustrating a diagram from the repair manual, with the system's display and spoken-language processing status confirmed at the top. Netscape Communicator browser window © 2000 Netscape Communications Corporation. Used with permission. Netscape Communications has not authorized, sponsored, endorsed, or approved this publication and is not responsible for its content.**



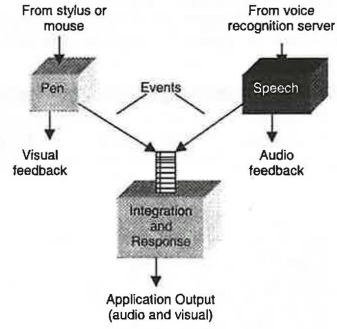
handwriting recognition in the fields of the order form. The platform is a Fujitsu Stylistic 1200 pen-based PC, with Windows 95 and a RangeLan2 PC card for wireless network connectivity.

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 this 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 speech and pen-based input, while still processing them as coordinated multimodal pieces.

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 identity, and objects are tagged in the parts diagrams when the application is configured. All words referring to parts are similarly tagged with this identity in the speech grammar. At runtime, this identity is included in the speech and pen input events when the object's name is spoken or

Amazon/VB Assets
Exhibit 1006
Page 42

Figure 14. **Data flow through the Portable Voice Assistant applet's three processing threads.**



the user's pen selects the image. 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 this prototype designed for general deployment, the speech vocabulary size ideally would be increased to handle more than 50 part names and 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 these prototypes' vocabulary is modest, the basic client–server architecture used for distributing speech recognition (see Figure 15) permits scaling to very large vocabularies.

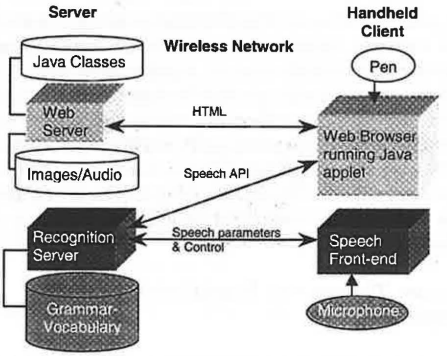
Our experience using the PVA indicates that a visually oriented Web-browsing and ordering application is very well suited to a multimodal interface. In addition, multimodal interaction is optimal during dynamically changing mobile environments, because the user can flexibly adapt her interaction in response 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 earlier have been built since the mid-1990s, and they still are research-level systems. How-

Amazon/VB Assets
Exhibit 1006
Page 43

Figure 15. **BBN's networked architecture for the Portable Voice Assistant, with speech recognition processing split up such that only feature extraction takes place on the handheld.**



ever, in some cases they have developed well beyond the prototype stage and are beginning to be integrated with a variety of other software at 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, 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, 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 need further work before new multimodal systems can be developed fully and eventually commercialized.

Amazon/VB Assets
Exhibit 1006
Page 44

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 participants, 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, Sandry, & Vidulich, 1983; Wickens, Vidulich, & Sandry-Garza, 1984, on Multiple Resource Theory). With respect to speech and gesture, linguistic and psychological literatures also have examined how people coordinate their spoken-language and natural manual gesturing during interpersonal communication (Kendon, 1980; Levelt, Richardson, & Heu, 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 et al., 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, 1999b; Oviatt & Kuhn, 1998), and this information has been used to establish basic NLP techniques specifically for multimodal systems.

Amazon/VB Assets
Exhibit 1006
Page 45

In the future, the relevant cognitive science literatures should be utilized more extensively as a basis for (a) hypothesizing about and examining multimodal human–computer communication, (b) spearheading parallel empirical work on other promising modality combinations besides speech and gesture (e.g., speech and gaze tracking), and (c) proposing innovative new system designs that are consistent with human information and sensory-processing advantages. Further empirical work is 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 is needed that specifically addresses issues such as (a) the behavioral features and automaticity of human communication modes, (b) 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 (c) basic intermodal coordination and synchronization issues.

5.2. New Multimodal Interface Concepts

Although the main multimodal subliteratures have focused on either speech or pen input and speech and lip movements (Rubin et al., 1998; Stork & Hennecke, 1995), 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 & 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 (e.g., speech, pen-based gestures, or other manual input), which the user *intends as a command issued to the system.* Although 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 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, Morimoto, & Ihde, 1999). This

Amazon/VB Assets
Exhibit 1006
Page 46