MULTIMODAL INTERFACES THAT PROCESS WHAT COMES NATURALLY

Sharon Oviatt and Philip Cohen

PUI

Using our highly skilled and coordinated communication patterns to control computers in a more transparent interface experience.

During multimodal communication, we speak, shift eye gaze, gesture, and move in a powerful flow of communication that bears little resemblance to the discrete keyboard and mouse clicks entered sequentially with a graphical user interface (GUI). A profound shift is now occurring toward embracing users' natural behavior as the center of the human-computer interface. Multimodal interfaces are being developed that permit our highly skilled and coordinated communicative behavior to control system interactions in a more transparent experience than ever before. Our voice, hands, and entire body, once augmented by sensors such as microphones and cameras, are becoming the ultimate transparent and mobile multimodal input devices.

The area of multimodal systems has expanded rapidly during the past five years. Since Bolt's [1] original "Put That There" concept demonstration, which processed speech and manual pointing during object manipulation, significant achievements have been made in developing more general multimodal systems. State-of-the-art multimodal speech and gesture systems now process complex gestural input other than pointing, and new systems have been extended to process different mode combinations—the most noteworthy being speech and pen input [9], and speech and lip movements [10]. As a foundation for advancing new multimodal systems, proactive empirical work has generated predictive information on human-computer multimodal interaction, which is being used to

guide the design of planned multimodal systems [7]. Major progress has occurred in both the hardware and software for component technologies like speech, pen, and vision. In addition, the basic architectural components and framework have become established for designing more general multimodal systems [3–5, 11]. Finally, real applications are being built that range from map-based and virtual reality systems for simulation and training, to field medic systems for mobile use in noisy environments, to Web-based transactions and standard text-editing applications [9]. All of these landmarks indicate progress toward building more general and robust multimodal systems, which will reshape daily computing tasks and have significant commercial impact in the future.

Here, we summarize the nature of new multimodal systems and how they work, with a focus on multimodal speech and pen-based input. To illustrate a multimodal speech and gesture architecture, the QuickSet system from the Oregon Graduate Institute of Science and Technology is introduced.

Accessibility for diverse users and usage contexts. Perhaps the most important reason for developing multimodal interfaces is their potential to greatly expand the accessibility of computing to diverse and nonspecialist users, and to promote new forms of computing not previously available [6, 9]. Since there can be large individual differences in people's abilities and preferences to use different modes of communication, multimodal interfaces will increase the accessibility of computing for users of different ages, skill levels, cognitive styles, sensory and motor impairments, native languages, or even temporary illnesses. This is because a multimodal interface permits users to exercise selection and control over how they interact with the computer. For example, a visually impaired user may prefer speech input, as may a manually impaired user with a repetitive stress injury or her arm in a cast. In contrast, a user with a hearing impairment, strong accent, or a cold may prefer pen input. Well before the keyboard is a practiced input device, a young preschooler could use either speech or pen-based drawing to control an educational application. A flexible multimodal interface also permits alternation of input modes, which prevents overuse and physical damage to any individual modality during extended periods of use. Just as the forearms can be damaged by repetitive stress when using a keyboard and mouse, the vocal cords also can be strained and eventually damaged by prolonged use of a speech system.

Multimodal systems that incorporate input modes like speech and pen also can facilitate new uses of computing—for example, in natural field settings and while mobile [9]. Any individual modality may be well suited for some tasks and environmental conditions, but less ideal or even inappropriate in others. A multimodal interface permits users to switch between modalities as needed during the continuously changing conditions of mobile use. Within a multimodal architecture, adaptive weighting of the input modes during environmental change can be performed to further enhance and stabilize the system's overall performance.

Performance stability and robustness. A second major reason for developing multimodal architectures is to improve the performance stability and robustness of recognition-based systems [6]. From a usability standpoint, multimodal systems offer a flexible interface in which people can exercise intelligence about how to use input modes effectively so that errors are avoided. To reap these error-handling advantages fully, a multimodal system must be designed so that the two input modes (for example, speech and pen) provide parallel or duplicate functionality, which means that users can accomplish their goals using either mode. A well-designed multimodal architecture also can support the mutual disambiguation of two input signals. For example, if a user says "ditches" but the speech recognizer confirms the singular "ditch" as its best guess, then parallel recognition of several graphic marks in pen input can result in recovery of the correct spoken plural interpretation. Technologists are just beginning to discover this kind of architectural pull-up can result in more accurate and stable system performance. In the future, we are increasingly likely to see promising but error-prone new media embedded within multimodal architectures in a way that harnesses and stabilizes them more effectively.

One of the most exciting recent discoveries is that multimodal systems demonstrate a relatively greater performance advantage precisely for those users and usage contexts in which unimodal systems fail. For example, recognition rates for unimodal spoken language systems are known to degrade rapidly for children or any type of nonnative accented speaker, and in noisy field environments or while users are mobile. However, recent research revealed a multimodal architecture can be designed that closes the recognition gap for these kinds of challenging users and usage contexts [6, 7]. As a result, next-generation multimodal systems may be capable of harnessing new media in a way that makes technology available to a broader range of everyday users and usage contexts than ever before.

Expressive power and efficiency. Systems that process multimodal input aim to give users a more powerful interface for accessing and manipulating information, such as increasingly sophisticated visualization and multimedia output capabilities [8]. In

contrast, interfaces that rely on keyboard and mouse input are limited or inappropriate for interacting with small mobile systems, virtual environments, and other new forms of computing. Since spoken and pen-based input are human language technologies, they can easily provide flexible descriptions of objects, events, spatial layouts, and their interrelation. For example, when using these modes together a person in one study said the following to place an open space park on a map:

User: [draws irregular area] "Open space."

In contrast, the same person composed the following lengthier and more disfluent utterance when only permitted to speak:

User: "Add an open space on the north lake to br—include the north lake part of the road and north."

This difference in linguistic efficiency and complexity is especially pronounced in applications that involve visual-spatial information [8].

In a recent study, multimodal interaction was demonstrated to be nine times faster when a user interacted with the pen/voice QuickSet system than when using a more familiar graphical interface for initializing simulation exercises [2]. This large efficiency advantage included the time required to correct recognition and manual errors in the two interfaces. When compared with rapid speech-only exchanges, multimodal pen/voice interaction also has resulted in a 10% increase in task completion time for spatial tasks [8]. In addition to these efficiency advantages, 90–100% of users prefer to interact multimodally during both spatial and nonspatial tasks [8].

How Multimodal Architectures Work

Multimodal systems are radically different than standard GUIs, largely because of the nature of human communication, and their basic architecture reflects these differences. Whereas input to GUIs is atomic and certain, machine perception of human input such as speech and gesture is uncertain, so any recognition-based system's interpretations are probabilistic. This means what were formerly basic events in a GUI, such as object selection, now are events that require recognition and are subject to misinterpretation. Secondly, whereas standard GUIs assume a sequence of discrete events, such as keyboard and mouse clicks, multimodal systems must process two or more continuous input streams that frequently are delivered simultaneously. The challenge for system developers is to create robust new time-sensitive architectures that support human communication patterns and performance, including processing users' parallel input and managing the uncertainty of recognition-based technologies.

One general approach to reducing or managing uncertainty is to build a system with at least two sources of information that can be fused. For example, various efforts are under way to improve speech recognition in noisy environments by using visually-derived information about the speaker's lip movements, called "visemes" [10], while interpreting "phonemes" or other features from the acoustic speech stream. Multimodal systems that interpret speech and lip motion integrate signals at the level of viseme and phoneme features that are closely relatedly temporally. Such architectures are based on machine learning of the viseme-phoneme correlations, using multiple hidden Markov models or temporal neural networks. This feature-level architectural approach generally is considered appropriate for modes that have similar time scales.

A second architectural approach, which is appropriate for the integration of modes like speech and gesture, involves fusing the semantic meanings of input signals. The two input signals do not need to occur simultaneously, and they can be recognized independently. This semantic fusion architectural approach requires less training data, and entails a simpler software development process [11]. As an illustration of the semantic fusion approach, we describe the QuickSet multimodal architecture and information processing flow.

QuickSet is an agent-based, collaborative multimodal system that runs on personal computers ranging from handheld to wall-sized [3]. The basic system has been developed in conjunction with a variety of map-based applications, including medical informatics, military simulation and training, 3D virtual-terrain visualization, and disaster management [3, 6]. QuickSet enables a user to create and position entities on a map or virtual terrain with speech, pen-based gestures, and/or direct manipulation. These entities then are used to populate a simulation or other map-based application. The user can create map-based objects by speaking their names and distinguishing characteristics, while simultaneously using a pen to designate information like location, number, and shape. For example, in the forest fire management scenario depicted in Figure 1, the location of a fire can be indicated by saying "burn line" while drawing its advancing edge. The user also can control objects in a simulation by specifying actions, such as "Jeep, follow this route," while drawing an evacuation route and the direction of vehicle movement. In addition to multimodal input, commands can be specified using either speech or gesture alone.

To interact with QuickSet, the user touches the screen to engage the microphone while speaking and

drawing. As each input signal arrives, its beginning and end are timestamped. These two signals are processed in parallel, as illustrated in Figure 2, with the recognition results generated by modality-specific understanding components. During recognition, these components produce a set of attribute/value meaning representations for each mode, called "feature structures." Structures of this type have been used extensively in the field of computational linguistics to encode lexical entries, grammar rules, and meaning representations. They resemble XML structures, but add the concept of logical variables derived from logic programming and a type hierarchy. The feature structures generated for an incoming signal provide alternative meaning hypotheses for that signal, each of which is assigned a probability estimate of correctness.

Figure 1. QuickSet running on tablet PC.

These feature structures are partial system interpretations, which then are passed to the multimodal integration component.

The multimodal integration stage is a three-step process that combines symbolic and statistical information to enhance system robustness. The integrator first uses signal timestamps to determine whether an incoming signal is unimodal or part of a multimodal command. To do this, QuickSet uses temporal constraints derived from empirical data, which has indicated that gestures precede or overlap with speech within a specific time threshold [7]. When a signal is potentially part of a multimodal command, QuickSet's integrator will attempt to combine the alternative feature structures with those from the other mode to form a complete semantic interpretation for the multimodal utterance.

After synchronizing signals, the multimodal integrator rapidly filters the alternative signal interpretations for semantic compatibility as well. Compatible or legal semantic combinations can either be stipulated linguistically, as is done in QuickSet, or derived from a corpus of prior interactions. After this semantic filtering process, the multimodal integrator fuses information from the two modes. To support semantic fusion, QuickSet uses a generalization of the term unification operation found in logic programming languages [5]. Unification is able to combine partial information from each of the two signals' interpretations, provided they are consistent. Incompatible information, such as from an incorrect recognition hypothesis, would be ruled out. Finally, if more than one multimodal interpretation is successfully unified, then the final interpretation is resolved from statistical rankings. These rankings are derived from weighted probability estimates for the spoken language and gestural pieces.

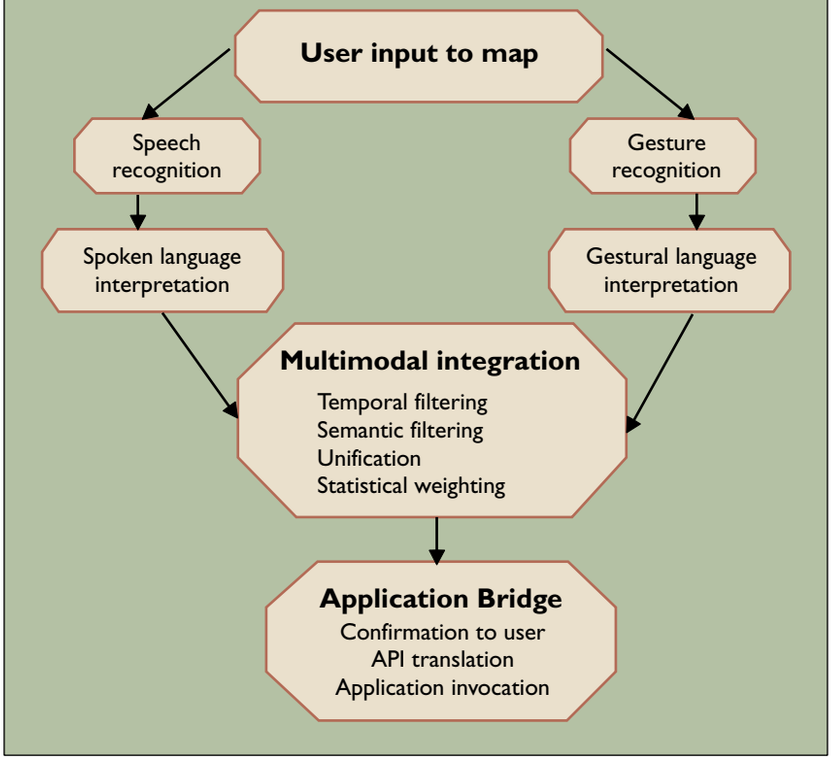
Since speech and gesture are highly interdependent, these final multimodal rankings are not calculated as a joint probability estimate. Instead, they are calculated as a linear weighting (with normalization) of the probability estimates for each signal. To estimate weighting coefficients, QuickSet uses a novel hierarchical recognition technique called Members-Teams-Committee (MTC) [11]. As illustrated in Figure 3, the MTC technique is comprised of a three-tiered divide-and-conquer recognition architecture with multiple members, multiple teams, and a committee. It uses a labeled corpus, with training proceeding in a bottom-up manner, layer-by-layer.

In the MTC approach, the members are the individual recognizers that provide an array of recognition results and probability estimates associated with input primitives (for example, stroke length). Member recognizers can contribute information to more than one team "leader," which then weights the reported scores. Each team can examine different subsets of data, and can apply a different weighting scheme. Finally, the committee weights the results derived from the various teams, and reports the final recognition results as a ranked list of alternative multimodal interpretations. The top-ranked interpretation is sent to the system's "application bridge" agent, which confirms the system's interpretation with the user and sends it to the appropriate backend application. In a recent evaluation, QuickSet's hybrid MTC architecture achieved over 95% correct recognition performance—or within 1.4% of the theoretical system upper bound [11].

There are numerous ways to realize this multimodal information processing flow as an architecture. One well-understood way is to pipeline various compo-

nents via remote procedure calls. However, this methodology can prove difficult if the system is heterogeneous. 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 used a multiagent architecture such as the Open Agent Architecture [4]. The components in such an architecture can be written in different languages and environments, although each component is wrapped by a layer of software that enables it to communicate via a standard language. The resulting component-with-communication-layer is called an agent. The agent communication language often uses message types derived from speech act theory, but they have been extended to handle asynchronous delivery, triggered responses, multicasting, and other concepts from distributed systems.

Figure 2. Information processing flow within QuickSet's multimodal architecture.



In some multiagent architectures, agents communicate directly with other components about which they have information. This design has the advantage of no intermediaries, although it can be brittle in the face of agent failure. As an alternative, many architectures have adopted a facilitated form of communication in which agents do not need to know with whom they are interacting. Rather, agents communicate through a known facilitator that routes messages to interested and capable receivers. The facilitator provides a place for new agents to connect at run time, enabling them to be discovered by other agents and incorporated into the functioning system. Figure 4 shows the same basic QuickSet components as Figure 2, but now arrayed around a central facilitator as part of a multiagent architecture. This architecture has proven flexible for adding and removing agents during operation. It also has supported distribution, multiuser collaboration, and cross-platform interoperation.

Multimodal speech and gesture systems developed since Bolt's "Put That There" prototype typically have been prone to a number of limitations, including:

- Functionality limited to simple point-and-speak integration patterns;
- Lack of a common meaning representation for interpreting two semantically rich input modes;
- Lack of a principled general approach to multimodal fusion; and
- Lack of a common reusable architecture for rapidly extending or building new multimodal systems.

In contrast, new systems like QuickSet handle complex gestural semantics and varied integration patterns. QuickSet's meaning representations and approach to semantic fusion are well-understood natural language processing techniques, which are used in concert with statistical learning algorithms. In addition, its multiagent architecture is extensible, and potentially could provide a common architecture for other types of multimodal systems.

In the future, other recognizers and information sources that provide probabilistic input (for instance, machine vision, dialogue context) could be incorporated within an MTC architecture. Such additional information could be used to derive more comprehensive, contextually sensitive, and robust multimodal interpretations. In addition, multimodal systems based on the MTC architecture could use empirical data to provide adaptive weightings that more successfully accommodate different types of user integration patterns [7], and moment-by-

moment changes in environmental noise levels.

Altering the Computing Experience

As the center of the human-computer interface shifts toward natural multimodal behavior, our extremely skilled and coordinated communication patterns will be used to control computers in a more transparent interface experience than ever before. Such interface designs will become more conversational in style, rather than limited to command and control, because many of the modes being processed are either language-oriented (speech, manual gestures, pen input) or involve communication broadly defined (gaze patterns, body movement).

As Turk and Robertson describe in their introductory remarks to this special section, some perceptual user interfaces (PUIs) can involve vision-based technology that unobtrusively monitors user behavior. Multimodal interfaces that combine two or more modes can incorporate an active input mode that the user intends as a command issued to the system, such as speech, pen-based gestures, or other manual input. They also can include a passive input mode that requires no explicit user command to the computer at all, such as vision-based tracking that senses a user's presence, gaze, and/or body position. While passive modes may be less obtrusive, active modes generally are more reliable indicators of user intent—which means that any unimodal recognition-based system can face a trade-off between the degree of obtrusive-

Affective Perception

Rosalind W. Picard

Imagine you have just logged into your new computer, and it is displaying some of its fancy features. It then begins asking you a series of questions. You are in a hurry to get to your email, but it pops up with yet another start-up window to set some option that is not necessary to configure now. You exhale, frown, mutter something under your breath, and proceed to type with a little more speed and intensity.

This scenario is one of many where a computer has caused an affective or emotional response. In this case, it was irritating its most important customer—the user. Despite the mantra of human-computer interaction—to design computers so as not to frustrate the user—computers still irritate, confuse, and annoy a great many people. We can all think of ways these interactions might be redesigned so that it would not frustrate us. Providing a delay start-up options button, for example, may be ideal for one person; however, that solution may confuse another person. There is rarely a one-size-fits-all solution for the growing variety of computer users and interactions.

People have skills for detecting when someone is annoyed or frustrated and for adapting to such affective cues. For example, if a human mentor is helping you with a task, then he or she can generally see when all is going well as opposed to when might be good to interrupt. Three factors are especially important: Perceiving the situation; perceiving affective expression; and knowing how interrupting at such a time was received previously. If, for example, a student is repeatedly doing something wrong (situation), but they are acting very curious and interested (affect), then the mentor might leave them alone. If however, their frustration is growing to the point of quitting (same situation, different affect), then it might be good to interrupt. The ultimate strategy involves more than affect perception, but affect perception is critical.

One of the goals of affective computing research is to give computers the ability to help communicate emotion—receiving and sending emotional cues [4]. If the interaction is primarily between you and the computer, then the goal of computer-emotion perception is to see whether such things pleased you, and thereby adjust its response more helpfully. This research involves comfortably sensing user's affective information, reasoning about the situation, and synthesizing a sensitive and respectful response.

A number of labs have built tools that enable affective cues to be directly communicated or indirectly. Emotional valence (liking or disliking) can be directly communicated by clicking on a thumbs-up or thumbs-down icon, or whacking physical icons of similar appearance, which may appear on the side of the computer or computing appliance. Intensity can be expressed via pressure applied to the mouse or to a physical icon. Valence, intensity, and other aspects of affective state can also be sensed indirectly from visual, auditory, or physiological cues.

Although facial expression and tone of voice may seem most natural for human affect recognition, it is important to respect the privacy wishes of users, and not to impose such technology. Several users have

ness and reliable functioning.

As vision-based technology and perceptual interfaces mature, however, some multimodal interfaces are forging a hybrid or blended interface style that combines both an active and passive mode. Blended multimodal interfaces can be "temporally cascaded" in the sense that one input mode precedes the other. Advance information arriving from the passively-tracked mode (eye gaze) typically is used to improve the multimodal system's prediction and interpretation of the active mode that follows (manual or speech input). An example of a cascaded active/passive interface is the IBM MAGIC system that passively tracks the user's gaze at a text field (right, left, above, or below the cursor location), while using this information to predict the direction of cursor movement and modulate a manual track pointer's physical resistance [12]. Among the goals of this particular multimodal interface are decreasing the user's manual fatigue and increasing input efficiency.

A blended interface potentially can perform more reliably than a pure passive-tracking system, because the active input mode is available to clarify user intentions that otherwise may be ambiguous. However, early information from the passive mode also can supply predictive power that enhances system robustness and delivers usability advantages, compared with just an active mode. As a result, in the future new blended multimodal interfaces may provide the user with greater transparency, better control, and a generally improved usability experience, while also supporting

expressed a preference for giving affective feedback via direct methods such as clicking on an icon or squeezing/hitting something. It would be ironic and irresponsible if affect-sensing technology, built to incorporate user feelings in the interaction, did not respect a user's feelings about how sensing was conducted.

One of the problems with so many smart features these days is they sense what you're doing but not how you're doing it. Popular word-processing software can sense you have misspelled a word, but not how you have tensed your muscles and grumbled as it keeps auto-correcting what you had, in fact, typed correctly. Even a dog senses how its master is responding and associates this feedback with its behavior. An infant senses how something is said long before he or she can understand what was said. To adapt behavior intelligently, living systems first perceive affective feedback.

Emotion plays a role in human perception. If subjects are asked to quickly jot words they hear, then they are more inclined to spell "presents" than "presence" if they are happy, and to spell "banned" than "band" if they are sad [2]. Similar results occur when subjects look at ambiguous facial expressions [1]. A variety of influences of emotion on perception have been described in [3].

Computers might potentially reason about the influence of mood on perception, to help them better predict what a person is likely to perceive. The computer that sees you are in a bad mood may predict that neutral language is likely to be perceived as negative, given that a negative mood may bias the ambiguous neutral stimulus negatively. The computer might thereby adjust its word choice in a way that would hardly be noticed, except that the communication would seem to have proceeded smoothly.

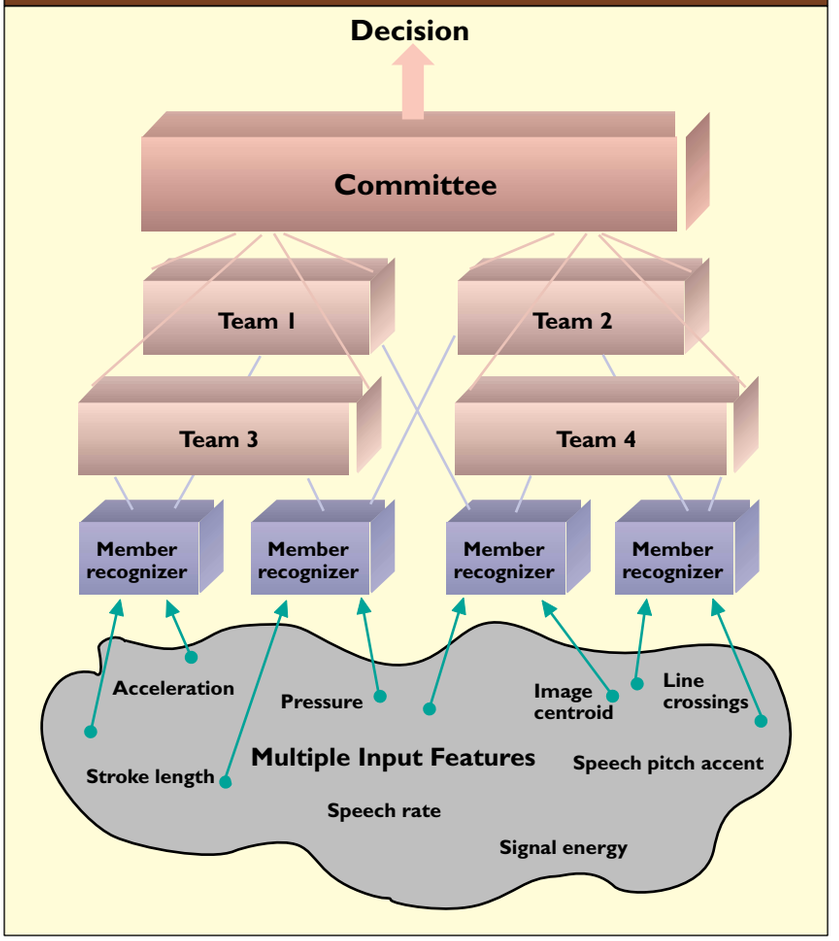
The ways in which affect is perceived, and in which it influences perception, are manifold and subtle. When they are missing, then human-human interaction is severely impaired. To the extent that human-machine interaction is natural and social, then machines will likely need affective skills. When the machine is being used as a hammer, then there is no need to clutter it with such features; however, when it is functioning as an assistant, helping you handle information overload and other complex tasks that require discerning and adapting to your individual goals, standards, and preferences, then affect perception will be a sign of intelligence. C

References

1. Bouhuys, A., Bloem, G.M., and Groothuis, T.G.G. Induction of depressed and elated mood by music influences the perception of facial emotional expression in healthy subjects. *J. Affective Disorders 33* (1995), 215–226.
2. Halberstadt, J.B., Niedenthal, P.M., and Kushner, J. Resolution of lexical ambiguity by emotional state. *Psychological Science 6*, 5 (Sept. 1995), 278–282.
3. Mayer, J.D., and Salovey, P. The intelligence of emotional intelligence. *Intelligence 17* (1993), 433–442.
4. Picard, R.W. *Affective Computing*. The MIT Press, Cambridge, MA, 1997.

Rosalind W. Picard (picard@media.mit.edu) is an associate professor at the MIT Media Lab in Cambridge, Mass. She founded and directs the Affective Computing Research Group.

Figure 3. Members-Team-Committee statistically based recognition approach, which is incorporated within QuickSet's hybrid architecture.



broader application functionality than a unimodal passive-monitoring PUI can alone.

Advancing the Field of Computer Science

Multimodal systems clearly will have an impact on various subareas of computer science. They initially will supplement, and eventually replace, the standard GUIs of today's computers for many applications. They also will become a focus for integrating many different capabilities from the field of artificial intelligence—such as machine vision, natural language processing, knowledge representation, reasoning, and machine learning. These capabilities are most likely to coalesce within a symbolic/statistical hybrid architecture based on a distributed multiagent framework.

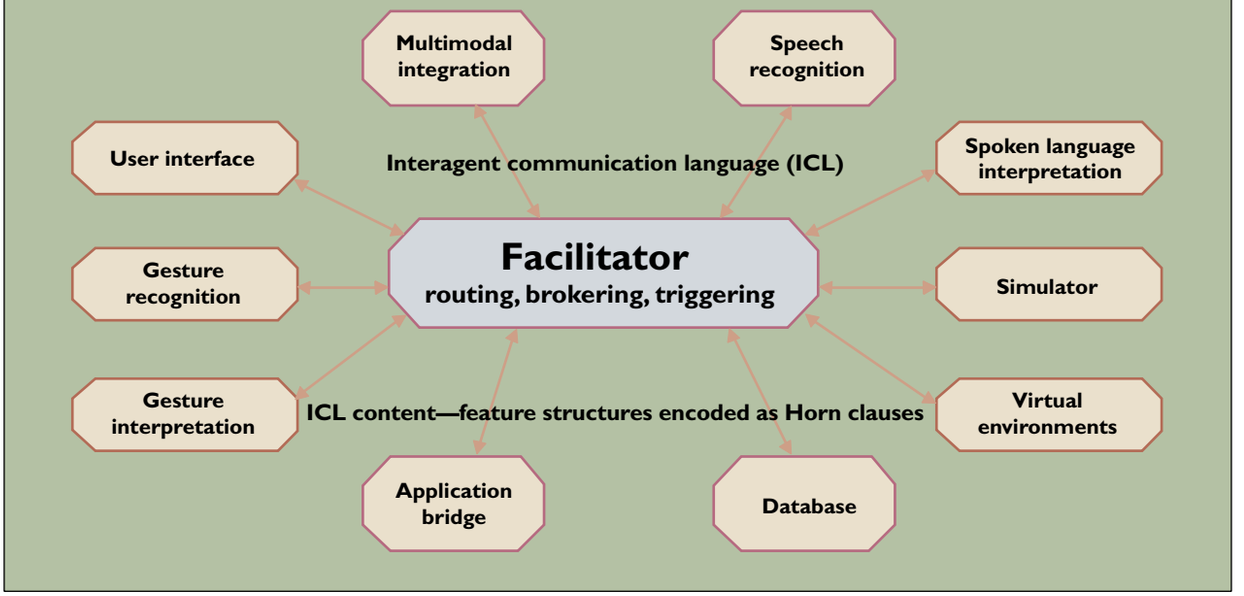
Multimodal interfaces also will advance in tandem with distributed computing, and will permit users to access and control a distributed information space in which the notions of applications, objects, and servers are hidden. Users should not need to know how to break down their information processing and communicative goals into the capabilities offered by individual software components. Instead, the "system" broadly conceived will decompose high-level goals into subgoals to be achieved by teams of autonomous agents that invoke services, retrieve information from databases or the Web, formulate alerts to watch for events of interest, and other functions.

In the area of mobile computing, multimodal interfaces will promote the broad use of small networked devices in extremely varied usage contexts, for both communications and computing purposes. Rather than simply offering a single-function information appliance, multimodal interfaces will promote the multifunctionality of small devices, in part due to the portability and expressive power of input modes like speech and pen.

Advancing the state-of-the-art of multimodal systems will require multidisciplinary expertise in a variety of areas beyond computer science—including speech and hearing science, perception and vision, linguistics, psychology, signal processing, pattern recognition, and statistics. The multidisciplinary nature of this new research agenda has several implications. To evolve successfully as a field, it means that computer science will need to become broader and more synthetic in its worldview, and to begin encouraging and rewarding researchers who successfully reach across the boundaries of their narrowly defined fields. It also means that any individual research group is unlikely to be able to conduct meaningful research across the entire spectrum. As a result, collaborative research and "community building" among multimodal researchers and sites will be critically needed to forge the necessary relations among those representing different key disciplines and component technologies.

In addition to cross-fertilization of ideas and perspectives among these diverse groups, there also is a critical need for cross-training of students and junior researchers. Like spoken language systems, multimodal technology does not fit neatly into a traditional

Figure 4. QuickSet's facilitated multiagent architecture.



academic departmental framework. To make the appropriate educational opportunities and resources available to future students, new multidisciplinary educational programs will need to be established that teach the relevant component technologies, scientific and engineering perspectives, research methods, and teamwork skills that will be needed to advance next-generation multimodal systems. Such programs could be fostered within existing academic departments, as well as new schools of information technology designed to promote intellectual ventures and training programs for the new millennium.

Conclusion

Multimodal systems are an emerging technology that offer expressive, transparent, efficient, robust, and mobile human-computer interaction. They also are strongly preferred by users for a variety of tasks and computing environments. Their increasingly sophisticated design and implementation are an important key to shifting the balance of human-computer interaction much closer to the human. C

References

1. Bolt, R.A. Put-that-there: Voice and gesture at the graphics interface. *ACM Computer Graphics 14*, 3 (1980), 262–270.
2. Cohen, P.R., Johnston, M., McGee, D., Oviatt, S.L., Clow, J., and Smith, I. The efficiency of multimodal interaction: A case study. In *Proceedings of the International Conference on Spoken Language Processing.* (Sydney, 1998), 249–252.
3. Cohen, P.R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. QuickSet: Multimodal interaction for distributed applications. In *Proceedings of the Fifth ACM International Multimedia Conference* (1997) ACM, NY, 31–40.
4. Cohen, P.R., Cheyer, A., Wang, M., and Baeg, S.C. An open agent architecture. *AAAI '94 Spring Symposium Series on Software Agents* AAAI, (Menlo Park, CA, 1994); reprinted in *Readings in Agents.* Morgan Kaufmann, 1997, 197–204.
5. Johnston, M., Cohen, P.R., McGee, D., Oviatt, S.L., Pittman, J.A., and Smith, I. Unification-based multimodal integration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, ACL* (New York, 1997), 281–288.
6. Oviatt, S.L. Mutual disambiguation of recognition errors in a multimodal architecture. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99)* ACM, NY, 576–583.
7. Oviatt, S.L. Ten myths of multimodal interaction. *Commun ACM 42,* 11, (Nov. 1999), 74–81.
8. Oviatt, S.L. Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction 12*, (1997), 93–129.
9. Oviatt, S.L., Cohen, P.R., Wu, L.,Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. and Ferro, D. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions for 2000 and beyond. *Human-Computer Interaction in the New Millennium.* J. Carroll, Ed., Addison-Wesley, Boston (in press).
10. Rubin, P., Vatikiotis-Bateson, E., and Benoit, C., Eds. Audio-visual speech processing. *Speech Communication 26,* (1998), 1–2.
11. Wu, L., Oviatt, S., and Cohen, P. Multimodal integration: A statistical view. *IEEE Transactions on Multimedia 1,* 4 (1999), 334–342.
12. Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99)* ACM, NY (1999), 246–253.

Sharon Oviatt (oviatt@cse.ogi.edu) and **Philip Cohen** (pcohen@cse.ogi.edu) are professors in the Department of Computer Science and Engineering at the Oregon Graduate Institute of Science and Technology (OGI), as well as co-directors of the Center for Human Computer Communication at OGI (www.cse.ogi.edu/CHCC).

This research was supported by Grant No. IRI-9530666 from the National Science Foundation and Special Extension for Creativity (SEC) Grant No. IIS-9530666 from NSF, Contracts DABT63-95-C-007 and N66001-99-D-8503 from DARPA's Information Technology and Information Systems offices, Grant No. N00014-95-1-1164 from ONR, and by grants, gifts, and equipment donations from Boeing, Intel, and Microsoft.