Confirmation in Multimodal Systems

David R. McGee, Philip R. Cohen and Sharon Oviatt
Center for Human-Computer Communication,
Department of Computer Science and Engineering
Oregon Graduate Institute
P.O. Box 91000, Portland, Oregon 97291-1000
{dmcgee, pcohen, oviatt}@cse.ogi.edu

ABSTRACT

Systems that attempt to understand natural human input make mistakes, even humans. However, humans avoid misunderstandings by confirming doubtful input. *Multimodal systems*—those that combine simultaneous input from more than one modality, for example speech and gesture—have historically been designed so that they either request confirmation of speech, their primary modality, or not at all. Instead, we experimented with delaying confirmation until after the speech and gesture were combined into a complete multimodal command. In controlled experiments, subjects achieved more commands per minute at a lower error rate when the system delayed confirmation, than compared to when subjects confirmed only speech. In addition, this style of *late confirmation* meets the user's expectation that confirmed commands should be executable.

KEYWORDS: multimodal, confirmation, uncertainty, disambiguation

"Mistakes are inevitable in dialog...In practice, conversation breaks down almost instantly in the absence of a facility to recognize and repair errors, ask clarification questions, give confirmation, and perform disambiguation. [1]"

INTRODUCTION

We claim that multimodal systems [2, 3] that issue commands based on speech and gesture input should not request confirmation of words or ink. Rather, these systems should, when there is doubt, request confirmation of their understanding of the combined meaning of each coordinated language act. The purpose of any confirmation act, after all, is to reach agreement on the overall meaning of each command. To test these claims we have extended our multimodal map system, *QuickSet* [4, 5], so that it can be tuned to request confirmation either before or after integration of modalities. Using QuickSet, we have conducted an empirical study that indicates agreement about the correctness of commands can be reached quicker if confirmation is delayed until after blending. This paper describes QuickSet, our experiences with it, an experiment that compares early and late confirmation strategies, the results of that experiment, and our conclusions.

Command-driven conversational systems need to identify hindrances to accurate understanding and execution of commands in order to avoid miscommunication. These hindrances can arise from at least three sources:

Uncertainty—lack of confidence in interpretation of the input,
Ambiguity—equally likely interpretations of input, and
Infeasibility—an inability to perform the command.

Suppose that we use a recognition system that interprets natural human input [6], that is capable of multimodal interaction [2, 3], and that will let users place simulated military units and related objects on a map. When we use this system, our words and stylus movements are simultaneously recognized, interpreted, and blended together. A user calls out the names of objects, such as "ROMEO ONE EAGLE," while marking the map with a gesture. If the system is confident of its recognition of the input, it might interpret this command in the following manner: a unit should be placed on the map at the specified location. Another equally likely interpretation, looking only at the results of speech recognition, might be to select an existing "ROMEO ONE EAGLE." Since this multimodal system is performing recognition, uncertainty inevitably exists in the recognizer's hypotheses. "ROMEO ONE EAGLE" may not be recognized with a high degree of confidence. It may not even be the most likely hypothesis.

One way to disambiguate the hypotheses is with the *multimodal language specification* itself, the way we allow modalities to combine. Since different modalities tend to capture complementary information [7-9], we can leverage this facility by combining ambiguous

spoken interpretations with disimilar gestures. For example, we might specify that selection gestures (circling) combine with the ambiguous speech from above to produce a selection command. Another way of disambiguating the spoken utterance is to enforce a precondition for the command: for example, for the selection command to be possible the object must already exist on the map. Thus, under such a precondition, if "ROMEO ONE EAGLE" is not already present on the map, the user cannot select it. We call these techniques *multimodal disambiguation* techniques.

Regardless, if a system receives input that it finds uncertain, ambiguous, or infeasible, or if its effect might be profound, risky, costly, or irreversible, it may want to verify its interpretation of the command with the user. For example, a system prepared to execute the command "DESTROY ALL DATA" should give the speaker a chance to change or correct the command. Otherwise, the cost of such errors is task-dependent and can be immeasurable [6, 10].

Therefore, we claim that conversational systems should be able to request the user to *confirm* the command, as humans tend to do [11-14]. Such confirmations are used "to achieve *common ground*" in human-human dialogue [15]. On their way to achieving common ground, participants attempt to minimize their *collaborative effort*, "the work that both do from the initiation of [a command] to its completion." [15] Herein we will further define collaborative effort in terms of work in a command-based collaborative dialogue, where an increase in the rate at which commands can be successfully performed corresponds to a reduction in the collaborative effort. We know that confirmations are an important way to reduce miscommunication [13, 16, 17], and thus collaborative effort. In fact, the more likely miscommunication, the more frequently people introduce confirmations [16, 17].

To ensure that common ground is achieved, miscommunication is avoided, and collaborative effort is reduced, system designers must determine when and how confirmations ought to be requested. Should a confirmation occur for each modality or should confirmation be delayed until the modalities have been blended? Choosing to confirm speech and gesture separately, or speech alone (as many contemporary multimodal systems do), might simplify the process of confirmation. For example, confirmations could be performed immediately after recognition of one or both modalities. However, we will show that collaborative effort can be reduced if multimodal systems delay confirmation until after blending.

1 MOTIVATION

Historically, multimodal systems have either not confirmed input [18-22] or confirmed only the primary modality of such systems—speech. This is reasonable, considering the evolution of multimodal systems from their speech-based roots. Observations of QuickSet prototypes last year, however, showed that simply confirming the results of speech recognition was often problematic—users had the expectation that whenever a command was confirmed, it would be executed. We observed that confirming speech prior to multimodal integration led to three possible cases where this expectation might not be met: ambiguous gestures, non-meaningful speech, and delayed confirmation.

The first problem with speech-only confirmation was that the gesture recognizer produced results that were often ambiguous. For example, recognition of the ink in Figure 1 could result in confusion. The arc (left) in the figure provides some semantic content, but it may be incomplete. The user may have been selecting something or she may have been creating an area, line, or route. On the other hand, the circle-like gesture (middle) might not be designating an area or specifying a selection; it might be indicating a circuitous route or line. Without more information from other modalities, it is difficult to guess the intentions behind these gestures.

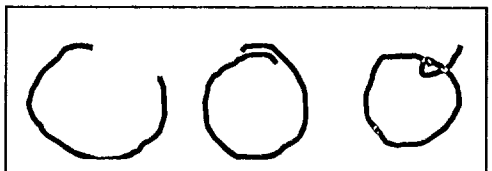

Figure 1. Ambiguous Gestures

Figure 1 demonstrates how, oftentimes, it is difficult to determine which interpretation is correct. Some gestures can be assumed to be fully specified by themselves (at right, an editor's mark meaning "cut"). However, most rely on complementary input for complete interpretation. If the gesture recognizer misinterprets the gesture, failure will not occur until integration. The speech hypothesis might not combine with any of the gesture hypotheses. Also, earlier versions of our speech recognition agent were limited to a single recognition hypothesis and one that might not even be syntactically

correct, in which case integration would always fail. Finally, the confirmation act itself could delay the arrival of speech into the process of multimodal integration. If the user chose to correct the speech recognition output or to delay confirmation for any other reason, integration itself could fail due to sensitivity in the multimodal architecture.

In all three cases, users were asked to confirm a command that could not be executed. An important lesson learned from these observations is that when confirming a command, users think they are giving approval; thus, they expect that the command can be executed without hindrance. Due to these early observations, we wished to determine whether delaying confirmation until after modalities have combined would enhance the human-computer dialogue in multimodal systems. Therefore, we hypothesize that *late-stage confirmations* will lead to three improvements in dialogue. First, because late-stage systems can be designed to present only feasible commands for confirmation, blended inputs that fail to produce a feasible command can be immediately flagged as a non-understanding and presented to the user as such, rather than as a possible command. Second, because of multimodal disambiguation, misunderstandings can be reduced, and therefore the number of conversational turns required to reach mutual understanding can be reduced as well. Finally, a reduction in turns combined with a reduction in time spent will lead to reducing the "collaborative effort" in the dialogue. To examine our hypotheses, we designed an experiment using QuickSet to determine if late-stage confirmations enhance human-computer conversational performance.

2 QUICKSET

This section describes QuickSet, a suite of agents for multimodal human-computer communication [4, 5].

2.1 A Multi-Agent Architecture

Underneath the QuickSet suite of agents lies a distributed, blackboard-based, multi-agent architecture based on the Open Agent Architecture[1] [23]. The blackboard acts as a repository of shared information and facilitator. The agents rely on it for brokering, message distribution, and notification.

[1] The Open Agent Architecture is a trademark of SRI International.

2.2 The QuickSet Agents

The following section briefly summarizes the responsibilities of each agent, their interaction, and the results of their computation.

2.2.1 User Interface

The user draws on and speaks to the interface (see Figure 2 for a snapshot of the interface) to place objects on the map, assign attributes and behaviors to them, and ask questions about them.

Figure 2. QuickSet Early Confirmation Mode

2.2.2 Gesture Recognition

The gesture recognition agent recognizes gestures from strokes drawn on the map. Along with coordinate values, each stroke from the user interface provides contextual information about objects touched or encircled by the stroke. Recognition results are an *n-best list* (top n-ranked) of interpretations. The interpretations are encoded as *typed feature structures* [5], which represent each of the potential semantic contributions of the gesture. This list is then passed to the *multimodal integrator.*

2.2.3 Speech Recognition

The Whisper speech recognition engine from Microsoft Corp. [24] drives the speech recognition agent. It offers speaker-independent, continuous recognition in close to real time. QuickSet relies upon a context-free domain grammar, specifically designed for each application, to constrain the speech recognizer. The speech recognizer

agent's output is also an n-best list of hypotheses and their probability estimates. These results are passed on for natural language interpretation.

2.2.4 Natural Language Interpretation

The natural language interpretation agent parses the output of the speech recognizer attempting to provide meaningful semantic interpretations based on a domain-specific grammar. This process may introduce further ambiguity; that is, more hypotheses. The results of parsing are, again, in the form of an n-best list of typed feature structures. When complete, the results of natural language interpretation are passed to the integrator for multimodal integration.

2.2.5 Multimodal Integration

The multimodal integration agent accepts typed feature structures from the gesture and natural language interpretation agents, and *unifies* them [5]. The process of integration ensures that modes combine according to a multimodal language specification, and that they meet certain multimodal timing and command-specific constraints. These constraints place limits on when different input can occur, thus reducing errors [7]. If after unification and constraint satisfaction, there is more than one completely specified command, the agent then computes the joint probabilities for each and passes the feature structure with the highest to the *bridge*. If, on the other hand, no completely specified command exists, a message is sent to the user interface, asking it to inform the user of the non-understanding.

2.2.6 Bridge to Application Systems

The bridge agent acts as a single message-based interface to domain applications. When it receives a feature structure, it sends a message to the appropriate applications, requesting that they execute the command.

3 CONFIRMATION STRATEGIES

QuickSet supports two modes of confirmation: early, which uses the speech recognition hypothesis; and late, which renders the confirmation act graphically using the entire integrated multimodal command. These two modes are detailed in the following subsections.

3.1 Early Confirmation

Under the *early confirmation* strategy (see Figure 3), speech and gesture are immediately passed to their respective recognizers (1a and 1b). Electronic ink is used for immediate visual feedback of the gesture input. The highest-scoring speech-recognition hypothesis is returned to the user interface and displayed for confirmation (2). Gesture recognition results are forwarded to the integrator after processing (4).

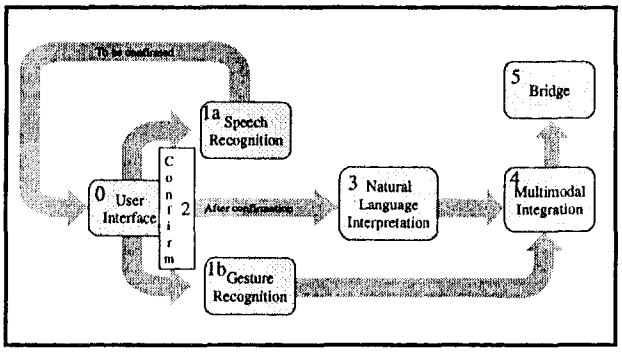

Figure 3. Early Confirmation Message Flow

After confirmation of the speech, QuickSet passes the selected sentence to the parser (3) and the process of integration follows (4). If, during confirmation, the system fails to present the correct spoken interpretation, users are given the choice of selecting it from a pop-up menu or respeaking the command (see Figure 2).

3.2 Late Confirmation

In order to meet the user's expectations, it was proposed that confirmations occur after integration of the multimodal inputs. Notice that in Figure 4, as opposed to Figure 3, no confirmation act impedes input as it progresses towards integration, thus eliminating the timing issues of prior QuickSet architectures.

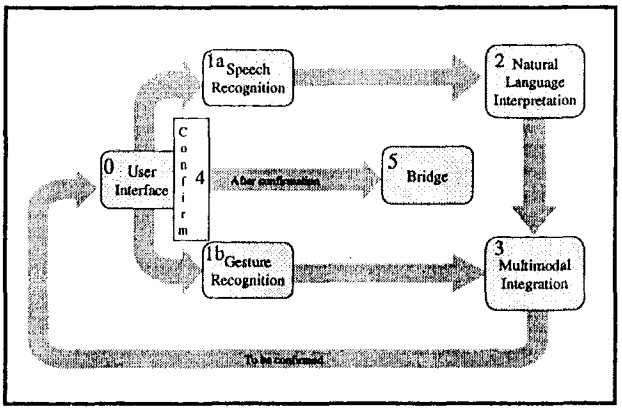

Figure 4. Late Confirmation Message Flow

Figure 5 is a snapshot of QuickSet in late confirmation mode. The user is indicating the placement of checkpoints on the terrain. She has just touched the map with her pen, while saying "YELLOW" to name the next checkpoint. In response, QuickSet has combined the gesture with the speech and graphically presented the

logical consequence of the command: a checkpoint icon (which looks like an upside-down pencil).

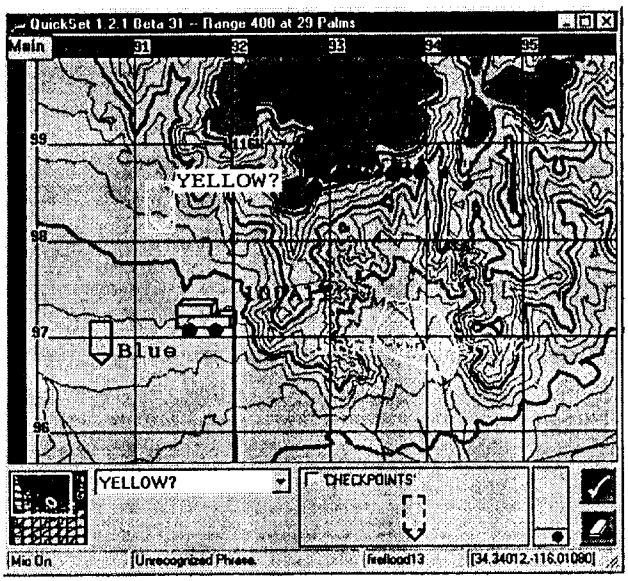

Figure 5. QuickSet in Late Confirmation Mode

To confirm or disconfirm an object in either mode, the user can push either the SEND (checkmark) or the ERASE (eraser) buttons, respectively. Alternatively, to confirm the command in late confirmation mode, the user can rely on *implicit confirmation*, wherein QuickSet treats non-contradiction as a confirmation [25-27]. In other words, if the user proceeds to the next command, she implicitly confirms the previous command.

4 EXPERIMENTAL METHOD

This section describes this experiment, its design, and how data were collected and evaluated.

4.1 Subjects, Tasks, and Procedure

Eight subjects, 2 male and 6 female adults, half with a computer science background and half without, were recruited from the OGI campus and asked to spend one hour using a prototypical system for disaster rescue planning.

During training, subjects received a set of written instructions that described how users could interact with the system. Before each task, subjects received oral instructions regarding how the system would request confirmations. The subjects were equipped with microphone and pen, and asked to perform 20 typical commands as practice prior to data collection. They performed these commands in one of the two confirmation modes. After they had completed either the flood or the fire scenario, the other scenario was introduced and the remaining confirmation mode was explained. At this time, the subject was given a chance to practice commands in the new confirmation mode, and then conclude the experiment.

4.2 Research Design and Data Capture

The research design was within-subjects with a single factor, confirmation mode, and repeated measures. Each of the eight subjects completed one fire-fighting and one flood-control rescue task, composed of approximately the same number and types of commands, for a strict recipe of about 50 multimodal commands. We counterbalanced the order of confirmation mode and task, resulting in four different task and confirmation mode orderings.

4.3 Transcript Preparation and Coding

The QuickSet user interface was videotaped and microphone input was recorded while each of the subjects interacted with the system. The following dependent measures were coded from the videotaped sessions: time to complete each task, and the number of commands and repairs.

4.3.1 Time to complete task

The total elapsed time in minutes and seconds taken to complete each task was measured: from the first contact of the pen on the interface until the task was complete.

4.3.2 Commands, repairs, turns

The number of commands attempted for each task was tabulated. Some subjects skipped commands, and most tended to add commands to each task, typically to navigate on the map (e.g., "PAN" and "ZOOM"). If the system misunderstood, the subjects were asked to attempt a command up to three times (repair), then proceed to the next one. Completely unsuccessful commands and the time spent on them, including repairs, were factored out of this study (1% of all commands). The number of turns to complete each task is the sum of the total number of commands attempted and any repairs.

4.3.3 Derived Measures

Several measures were derived from the dependent measures. *Turns* per *command* (tpc) describes how many turns it takes to successfully complete a command. *Turns* per *minute* (tpm) measures the speed with which the user interacts. A multimodal error rate was calculated based on how often repairs were

necessary. *Commands* per *minute* (cpm) represents the rate at which the subject is able to issue successful commands, estimating the collaborative effort.

5 RESULTS

	Means		One-tailed t-test (df=7)
	Early	Late	
Time(min.)	13.5	10.7	$t = 2.802, p < 0.011$
tpc	1.2	1.1	$t = 1.759, p < 0.061$
tpm	4.5	5.3	$t = -4.00, p < 0.003$
Error rate	20%	14%	$t = 1.90, p < 0.05$
cpm	3.8	4.8	$t = -3.915, p < 0.003$

These results show that when comparing late with early confirmation: 1) subjects complete commands in fewer turns (the error rate and tpc are reduced, resulting in a 30% error reduction); 2) they complete turns at a faster rate (tpm is increased by 21%); and 3) they complete more commands in less time (cpm is increased by 26%). These results confirm all of our predictions.

6 DISCUSSION

There are two likely reasons why late confirmation outperforms early confirmation: implicit confirmation and multimodal disambiguation. Heisterkamp theorized that implicit confirmation could reduce the number of turns in dialogue [25]. Rudnicky proved in a speech-only digit-entry system that implicit confirmation improved throughput when compared to explicit confirmation [27], and our results confirm their findings. Lavie and colleagues have shown the usefulness of *late-stage disambiguation*, during which speech-understanding systems pass multiple interpretations through the system, using context in the final stages of processing to disambiguate the recognition hypotheses [28]. However, we have demonstrated and empirically shown the advantage in combining these two strategies in a multimodal system.

It can be argued that implicit confirmation is equivalent to being able to undo the last command, as some multimodal systems allow [3]. However, commands that are infeasible, profound, risky, costly, or irreversible are difficult to undo. For this reason, we argue that implicit confirmation is often superior to the option of undoing the previous command. Implicit confirmation, when combined with late confirmation, contributes to a smoother, faster, and more accurate collaboration between human and computer.

7 CONCLUSIONS

We have developed a system that meets the following expectation: when the proposition being confirmed is a command, it should be one that the system believes can be executed. To meet this expectation and increase the conversational performance of multimodal systems, we have argued that confirmations should occur late in the system's understanding process, at a point after blending has enhanced its understanding. This research has compared two strategies: one in which confirmation is performed immediately after speech recognition, and one in which it is delayed until after multimodal integration. The comparison shows that late confirmation reduces the time to perform map manipulation tasks with a multimodal interface. Users can interact faster and complete commands in fewer turns, leading to a reduction in collaborative effort.

A direction for future research is to adopt a strategy for determining whether a confirmation is necessary [29, 30], rather than confirming every utterance, and measuring this strategy's effectiveness.

ACKNOWLEDGEMENTS

This work is supported in part by the Information Technology and Information Systems offices of DARPA under contract number DABT63-95-C-007, and in part by ONR grant number N00014-95-1-1164. It has been done in collaboration with the US Navy's NCCOSC RDT&E Division (NRaD). Thanks to the faculty, staff, and students who contributed to this research, including Joshua Clow, Peter Heeman, Michael Johnston, Ira Smith, Stephen Sutton, and Karen Ward. Special thanks to Donald Hanley for his insightful editorial comment and friendship. Finally, sincere thanks to the people who volunteered to participate as subjects in this research.

REFERENCES

[1] D. Perlis and K. Purang, "Conversational adequacy: Mistakes are the essence," in *Proceedings of Workshop on Detecting, Repairing, and Preventing Human-Machine Miscommunication, AAAI'96*, 1996.

[2] R. Bolt, "Put-That-There: Voice and gesture at the graphics interface," *Computer Graphics*, vol. 14, pp. 262-270, 1980.

[3] M. T. Vo and C. Wood, "Building an Application Framework for Speech and Pen Input Integration in Multimodal Learning Interfaces," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'96*, Atlanta, GA, 1996.

[4] P. R. Cohen, M. Johnston, D. McGee, I. Smith, J. Pittman, L. Chen, and J. Clow, "Multimodal interaction for distributed interactive simulation," in *Proceedings of Innovative Applications of Artificial Intelligence Conference, IAAI'97*, Menlo Park, CA, 1997.

[5] M. Johnston, P. R. Cohen, D. McGee, S. L. Oviatt, J. A. Pittman, and I. Smith, "Unification-based multimodal integration," in *Proceedings of 35th Annual Meeting of the Association for Computational Linguistics, ACL'97*, Madrid, Spain, 1997.

[6] J. R. Rhyne and C. G. Wolf, "Chapter 7: Recognition-based user interfaces," in *Advances in Human-Computer Interaction*, vol. 4, H. R. Hartson and D. Hix, Eds., pp. 191-250, 1992.

[7] S. Oviatt, A. DeAngeli, and K. Kuhn, "Integration and synchronization of input modes during multimodal human-computer interaction," in *Proceedings of Conference on Human Factors in Computing Systems, CHI'97*, pp. 415-422, Atlanta, GA, 1997.

[8] P. Lefebvre, G. Duncan, and F. Poirier, "Speaking with computers: A multimodal approach," in *Proceedings of EUROSPEECH'93 Conference*, pp. 1665-1668, Berlin, Germany, 1993.

[9] P. Morin and J. Junqua, "Habitable interaction in goal-oriented multimodal dialogue systems," in *Proceedings of EUROSPEECH'93 Conference*, pp. 1669-1672, Berlin, Germany, 1993.

[10] L. Hirschman and C. Pao, "The cost of errors in a spoken language system," in *Proceedings of EUROSPEECH'93 Conference*, pp. 1419-1422, Berlin, Germany, 1993.

[11] H. Clark and D. Wilkes-Gibbs, "Referring as a collaborative process," *Cognition*, vol. 13, pp. 259-294, 1986.

[12] P. R. Cohen and H. J. Levesque, "Confirmations and joint action," in *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 951-957, 1991.

[13] D. G. Novick and S. Sutton, "An empirical model of acknowledgment for spoken-language systems," in *Proceedings of 32nd Annual Meeting of the Association for Computational Linguistics, ACL'94*, pp. 96-101, Las Cruces, New Mexico, 1994.

[14] D. Traum, "A Computational Theory of Grounding in Natural Language Conversation," Computer Science Department, University of Rochester, Rochester, NY, Ph.D. 1994.

[15] H. H. Clark and E. F. Schaefer, "Contributing to discourse," *Cognitive Science*, vol. 13, pp. 259-294, 1989.

[16] S. L. Oviatt, P. R. Cohen, and A. M. Podlozny, "Spoken language and performance during interpretation," in *Proceedings of International Conference on Spoken Language Processing, ICSLP'90*, pp. 1305-1308, Kobe, Japan, 1990.

[17] S. L. Oviatt and P. R. Cohen, "Spoken language in interpreted telephone dialogues," *Computer Speech and Language*, vol. 6, pp. 277-302, 1992.

[18] G. Ferguson, J. Allen, and B. Miller, "The design and implementation of the TRAINS-96 system: A prototype mixed-initiative planning assistant," University of Rochester, Rochester, NY, TRAINS Technical Note 96-5, October 1996 1996.

[19] G. Ferguson, J. Allen, and B. Miller, "TRAINS-95: Towards a mixed-initiative planning assistant," in *Proceedings of Third Conference on Artificial Intelligence Planning Systems, AIPS'96*, pp. 70-77, 1996.

[20] D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J. Polifroni, S. Seneff, and V. Zue, "GALAXY: A Human-Language Interface to On-Line Travel Information," in *Proceedings of International Conference on Spoken Language Processing, ICSLP '94*, pp. 707-710, Yokohama, Japan, 1994.

[21] R. Lau, G. Flammia, C. Pao, and V. Zue, "WebGALAXY: Spoken language access to information space from your favorite browser," Massachusetts Institute of Technology, Cambridge, MA, URL http://www.sls.lcs.mit.edu/SLSPublications.html, December 1997 1997.

[22] V. Zue, "Navigating the information superhighway using spoken language interfaces," *IEEE Expert*, pp. 39-43, 1995.

[23] P. R. Cohen, A. Cheyer, M. Wang, and S. C. Baeg, "An open agent architecture," in *Proceedings of AAAI 1994 Spring Symposium on Software Agents*, pp. 1-8, 1994.

[24] X. Huang, A. Acero, F. Alleva, M.-Y. Hwang, L. Jiang, and M. Mahajan, "Microsoft Windows Highly Intelligent Speech Recognizer: Whisper," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'95*, 1995.

[25] P. Heisterkamp, "Ambiguity and uncertainty in spoken dialogue," in *Proceedings of EUROSPEECH'93 Conference*, pp. 1657-1660, Berlin, Germany, 1993.

[26] Y. Takebayashi, "Chapter 14: Integration of understanding and synthesis functions for multimedia interfaces," in *Multimedia interface design*, M. M. Blattner and R. B. Dannenberg, Eds. New York, NY: ACM Press, pp. 233-256, 1992.

[27] A. I. Rudnicky and A. G. Hauptmann, "Chapter 10: Multimodal interaction in speech systems," in *Multimedia Interface Design*, M. M. Blattner and R. B. Dannenberg, Eds. New York, NY: ACM Press, pp. 147-171, 1992.

[28] A. Lavie, L. Levin, Y. Qu, A. Waibel, and D. Gates, "Dialogue processing in a conversational speech translation system," in *Proceedings of International Conference on Spoken Language Processing, ICSLP'96*, pp. 554-557, 1996.

[29] R. W. Smith, "An evaluation of strategies for selective utterance verification for spoken natural language dialog," in *Proceedings of Fifth Conference on Applied Natural Language Processing, ANLP'96*, pp. 41-48, 1996.

[30] Y. Niimi and Y. Kobayashi, "A dialog control strategy based on the reliability of speech recognition," in *Proceedings of International Conference on Spoken Language Processing, ICSLP'96*, pp. 534-537, 1996.