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

Recognition systems attempt to understand natural human input. No matter how sophisticated the recognition system may be, though—mistakes are inevitable. Even humans, although superior recognizers, make mistakes, but they attempt to avoid misunderstandings created by mistakes in two ways: by acknowledging what others are saying and by requesting confirmation of what was said when there is doubt. Like human beings, multimodal systems blend input from several recognizers, enhancing the system's understanding of the input. We found that applying confirmations after this blending, at a late-stage in the understanding process, noticeably reduces the collaborative effort in multimodal human-computer dialogue. In addition, we have demonstrated that systems using late-stage confirmation meet 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

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
Unfeasibility—an inability to perform the command

Suppose that we use a recognition system, such as those described by Rhyne and Wolf [2], capable of multimodal interaction [3, 4] that will let users place objects on a map. When we use this system, our spoken words and pen 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 point. One interpretation of this interaction might be that an object should be placed on the map at the specified location. However, since this multimodal system is performing recognition, uncertainty inevitably exists in the recognizer's hypotheses, which are only estimations of the likelihood that any particular interpretation is correct. In other words, "`ROMEO ONE EAGLE`" may not be recognized with a high degree of confidence. It may not even be the most likely hypothesis, since many are available from the recognizer. Likewise, ambiguity is present in our example utterance. Specifying the location of a new object or selecting existing objects might use the same linguistic components. "`ROMEO ONE EAGLE`" could be either a request to select that object or a command to create an object with that name.

One way to disambiguate the hypotheses is in the *multimodal language specification* itself, the way modalities combine. Since different modalities tend to capture complementary information [5-7], we can leverage this facility by combining the ambiguous spoken interpretations with different gestures. For example, we might specify that selection gestures combine with the ambiguous speech to produce a selection command. Another way of disambiguating the spoken utterance is to enforce a precondition for the command: for example, the object must already exist on the map for the selection command to be possible. Thus, under such a precondition, if "`ROMEO ONE EAGLE`" is not already present on the map, then the user simply cannot select it. We call these techniques *multimodal disambiguation*.

If a system receives input that it finds uncertain, ambiguous, or infeasible, it may want to verify with the user its interpretation of the command. Another reason to verify the command is that its effect might be profound, risky, costly, or irreversible. 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 [2, 8].

We claim that human-machine communication should attempt to emulate human-human conversation in many ways. Namely, conversational systems should be able to request the user to *confirm* the command [9-12]. Such confirmations are used "to achieve common ground" in human-human dialogue [13]. 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. [13]" Confirmations are an important way to reduce miscommunication [11, 14, 15], and thus collaborative effort. In fact, the more likely miscommunication, the more frequently people introduce confirmations [14, 15].

In multimodal systems, to assure 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 blending of modalities has occurred? 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, without waiting until blending is complete. However, rather than confirming words or ink, we claim that the system should instead request confirmation of its understanding of the combined meaning of the coordinated language act. The purpose of any confirmation act, after all, is to agree on some level of overall meaning.

To test these claims we have extended our multimodal map system, QuickSet, 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 validates our claim. This paper describes QuickSet; our experiences with it, which motivated this research; an experiment that compared early and late confirmation strategies; the results of that experiment; and our conclusions.

1 QUICKSET

This section describes QuickSet, a suite of agents for multimodal human-computer communication [16, 17].

1.1 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] [18]. The blackboard acts as a repository of shared information and as a facilitator. The agents rely on this facilitator for brokering, message distribution, and notification.

1.2 The QuickSet Agents

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

1.2.1 User Interface

The user draws on and speaks to the interface (see Figure 1 for a snapshot of the interface) to place objects on the map, assign attributes and behaviors to them, and ask questions about them. More on the QuickSet user interface is described in Section 2.

Figure 1. QuickSet Early Confirmation Mode

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

1.2.2 Gesture Recognition

The gesture recognition agent recognizes gestures from strokes drawn on the map. Along with the coordinate values, each stroke from the user interface also provides contextual information about objects touched or encircled by the stroke. Recognition results are an *n-best list* (top n-ranked) of interpretations and an associated probability estimate for each interpretation. The interpretations are encoded as *typed feature structures* [17], which represent semantic contributions. This list is then passed to the *multimodal integrator*.

1.2.3 Speech Recognition

The Microsoft Whisper speech recognition engine drives the automatic speech recognition (ASR) agent used here, offering a combination of relevant features, all in close to real time: speaker-independent, continuous recognition, as well as multiple hypotheses and their probability estimates. The speech recognizer's output, like the gesture recognizer's, is an n-best list comprised of probability estimates and hypotheses. These results are passed on for natural language interpretation.

1.2.4 Natural Language Interpretation (Parser)

The natural language interpretation agent parses the output of the ASR attempting to provide proper semantic interpretations. This process may introduce further ambiguity; that is, more hypotheses. 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.

1.2.5 Multimodal Integration (Integrator)

The multimodal integration agent accepts typed feature structures from the gesture and parser agents, and *unifies* them [17]. The process of integration ensures that modes combine according to a language specification, and that they meet certain multimodal timing constraints. These constraints place limits on when different input can occur, reducing errors [5]. Integrations that do not result in a completely specified command are ignored. The agent then examines the joint probabilities for any remaining command and passes the feature structure with the highest joint probability to the *bridge*. If none exists, a message is sent to the user interface, asking it to inform the user of the non-understanding.

1.2.6 Bridge to Application Systems

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

2 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.

2.1 Early Confirmation

Under the *early confirmation* strategy (see Figure 2), the highest-scoring speech-recognition hypothesis (1a) is immediately passed to the user interface and displayed for confirmation (2). Electronic ink (the red circle next to the flood zone in Figure 1) is used for immediate visual feedback of the gesture input, while the input itself is passed on to the gesture recognizer without confirmation (1b). Gesture recognition results are forwarded to the integrator after processing (4).

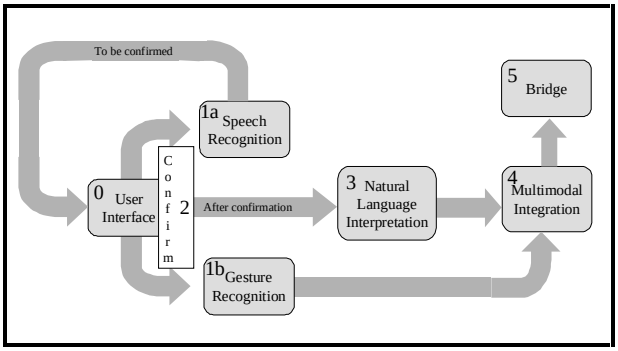

Figure 2. Early Confirmation Message Flow

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 1).

2.2 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 3, as opposed to Figure 2, no confirmation act impedes input as it progresses towards integration, thereby eliminating the

timing problems in the early QuickSet architecture (see discussion, Section 6).

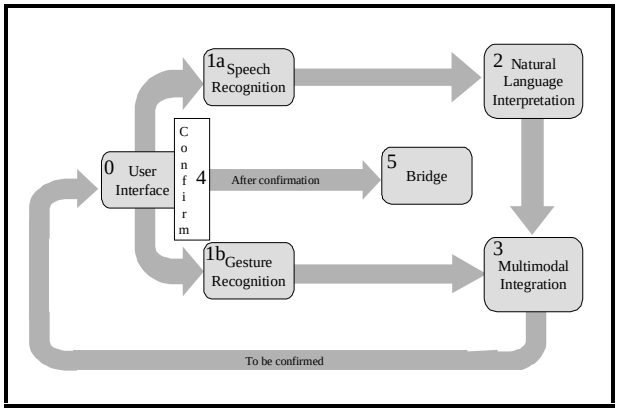

Figure 3. Late Confirmation Message Flow

Figure 4 is a snapshot of QuickSet in late confirmation mode. The user is indicating the placement of checkpoints on the terrain. He has just touched the map with his 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, in dashed lines, and boxed about in solid yellow lines. If the user confirms the integrated command, the solid lines disappear and the dashed lines solidify.

Figure 4. QuickSet in Late Confirmation Mode

To confirm or disconfirm an object in either confirmation 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 simply rely on *implicit confirmation*, wherein QuickSet treats non-contradiction as a confirmation [19-21]. In other words, if the user proceeds to the next command, she confirms the previous command.

3 MOTIVATION

Historically, multimodal systems have either not confirmed input 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 5 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.

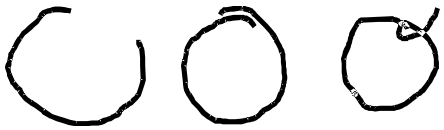

Figure 5. Ambiguous Gestures

The figure demonstrates how, oftentimes, the gesture agent alone cannot determine which of the interpretations is correct. Though some gestures are fully specified by themselves (at right, an editor's mark meaning "cut"), most rely on complementary input for complete interpretation. If the gesture recognizer misinterprets the gesture, failure will not occur until integration. In fact, the speech hypothesis might not combine with any of the gestures recognized. The second case where user's expectation may not be met: earlier versions of the speech recognition agent were limited to a single interpretation of the utterance and one that might not even be syntactically

correct, in which case integration would always fail. Finally, performing the confirmation act itself could delay the arrival of speech input into the process of multimodal integration. If the user chose to correct an error in the speech recognition output using another modality or to delay confirmation for any other reason, integration itself could fail due to the multimodal architecture's time sensitivity.

In all three cases, users were asked to confirm a command that might never be executed. An important lesson learned from these observations is that when confirming a command, users think they are giving approval; thus, they expect the command can be executed without hindrance. If it cannot be, a display of non-understanding or a clarification may be better than a confirmation.

Consequently, we wished to determine whether delaying confirmation until after modalities have combined would enhance the human-computer dialogue in multimodal systems. We hypothesize that *late-stage confirmations*, hereafter called *late confirmations*, will lead to three improvements in the 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 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 to determine if late-stage confirmations enhance human-computer conversational performance as predicted.

4 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 outlined how users should interact with the system, and before each task, subjects received oral instructions regarding how the system would request confirmations. As practice the subjects were equipped with microphone and pen, and asked to perform 20 typical commands prior to data collection.

4.2 Research Design and Data Capture

The research design was within-subjects with a single factor and repeated measures. The independent factor of interest was confirmation mode, either late or early. 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 planning commands. While performing one of the tasks, the subject used early confirmation; on the other, late confirmation. Order was counterbalanced for both 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[2]). The number of turns to complete each task is a composite of the total number of commands attempted and any repairs; that is

 # Turns = # Commands + # 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. C*ommands* 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 [19]. Rudnicky proved in a speech-only digit-entry system that implicit confirmation improved throughput when compared to explicit confirmation [21], 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 [22]. 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 [4]. However, commands that are infeasible, profound, risky, costly, or irreversible will be difficult, if not impossible, to undo. For this reason, we argue that implicit confirmation is often superior to the option of undoing the previous command. Implicit confirmation, after all, does not deny the user the option of undoing the prior command; when combined with late confirmation, it contributes to a smoother, faster, and overall 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 [23, 24], 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

[2] A preliminary inquiry shows that this 1% would not significantly influence our findings. However, an investigation of more general hypotheses would demand an evaluation of the influence of this 1% on any predictions.

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] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

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

[11] 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.

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

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.