3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

A USER-INITIATED DIALOGUE MODEL AND ITS IMPLEMENTATION FOR SPONTANEOUS HUMAN-COMPUTER INTERACTION

Hiroshi Kanazawa, Shigenobu Seto, Hideki Hashimoto, Hideaki Shinchi and Yoichi Takebayashi

Toshiba Corporation, Research & Development Center
Saiwai-ku, Kawasaki, 210 Japan

ABSTRACT

This paper describes a user-initiated dialogue model and its implementation on TOSBURG II, a real-time task-oriented spontaneous speech dialogue system, for achieving robust human-computer interaction. In order to make TOSBURG II user-initiative and flexible, we designed it on a semantic interpretation and user-initiated dialogue management basis. To implement the user-initiated dialogue model on a robust user-unspecified system, we have introduced the ATN (Augmented Transition Network) and represented the model, including eight user states and 19 system states, within the ATN framework. Through dialogue experiments on unspecified users, TOSBURG II has shown to be highly robust and friendly. It is confirmed that our dialogue model facilitates user-initiated spontaneous human-computer interaction.

1. INTRODUCTION

Humans conduct many kinds of dialogue between each other using spontaneous speech depending on situations and purposes. Therefore, dialogue management relevant to dialogue situations is as important as spontaneous speech understanding for natural human-computer interaction[1]~[3]. This suggests that the dialogue manager should handle inevitable ambiguities and errors in the speech understanding process and guide users toward the dialogue goal. Many conventional human-computer dialogue systems employ a computer-initiated dialogue management, in which users must follow request messages from the computer. To realize spontaneous interaction, we have proposed a user-initiated dialogue model and implemented it to the real-time spontaneous dialogue system TOSBURG II[4]. This paper first presents our approach to the user-initiated dialogue system. Next, the system's features are described. Third, processes of dialogue management and implementation are given. Finally, experimental results obtained using a real-time system are shown.

2. USER-INITIATED SPONTANEOUS SPEECH DIALOGUE SYSTEM

2.1 Approach

With the development of graphical user interfaces, the type of dialogue with computers has changed from "Remember and Type" to "See and Point". Thanks to the recent advances in technologies, speech recognition is now easily implemented on computers and an "Ask and Tell" interface is expected as the next generation interface[5]. Taking these points into account, we designed human-computer interaction based on an "Ask and Tell" interface and developed a user-initiated spontaneous speech dialogue system without imposing any restrictions on the user's utterances, in order to enhance the naturalness and friendliness of speech media.

2.2 Spontaneous Dialogue

Considering the approach described above, we have constructed a task-oriented speech dialogue system based on spontaneous speech understanding and response generation (TOSBURG II), as shown in Fig.1. We have employed a keyword-based approach to understanding spontaneous speech, which may include unintentional and unpredictable utterances. The meaning of an utterance can be extracted by combining keyword-spotting with keyword lattice parsing. The parser supplies likelihood values to the associated semantic utterance representation, which is expressed by a frame-type knowledge representation including an act slot and item slots. Semantic information on acts such as making orders and cancelling is represented in the act slot; the item slots contain semantic information on ordered items including food items, size and number[6].

For spontaneous interaction, we employ a user-initiated dialogue manager and speech response canceller. The system allows users' interruption using synthetic speech cancellation by the LMS algorithm[7]. Our dialogue manager realizes user-initiated dialogue by referring to the dialogue history and context. If the system's interpretation is ambiguous, it invites the user to verbally confirm it. There is no need for users to follow the system's instructions; users can speak spontaneously.

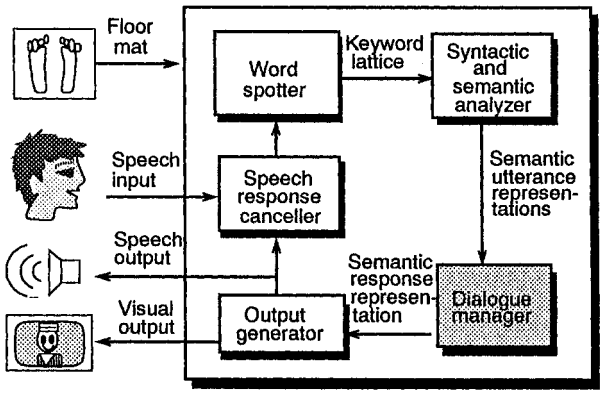

Fig.1 Configuration of speech dialogue system TOSBURG II

3. USER-INITIATED DIALOGUE MANAGEMENT

This section discusses our dialogue model and dialogue management.

3.1 Dialogue Model

In a system-initiated dialogue, most tasks are performed by following the system's instructions. For spontaneous speech dialogue, it is necessary to deal with unintentional utterances, changes in word order, ellipses, and other phenomena of spontaneous speech and remove any restrictions on the user's utterances as much as possible. Furthermore, since the user's internal state changes as the dialogue progresses, it is also necessary to interpret the user's utterances flexibly. Our system takes into its model the transition between user states and system states, as shown in Fig.2.

First, the system outputs greetings like "Welcome, may I take your order?" and then switches to the user's initial state. Next, after the user makes his order, it switches to the system's dialogue continuation state. Then the system modifies the dialogue history according to the semantic utterance representation, generates confirmation query, and requests responses. In this way, the user's utterance is interpreted while predicting the user's internal state from the system's replies. Then, after the system determines that all orders have been confirmed, it switches to the end state, outputs the message "Thank you very much" and terminates the dialogue.

3.2 Dialogue Topic Management

To deal with ambiguities and recognition errors, our system conducts dialogue by generating confirmational responses to the user. In user-initiated dialogue, the user does not necessarily respond as the system expects. For example, when the system asks for confirmation on the number of items ordered, the user may respond with additional orders or make corrections. To solve this problem, our system employs an order table consisting of the stack of both confirmed and unconfirmed orders. It controls the dialogue topics using the stack operation, as shown in Fig.3. When the user requests additional orders or makes corrections, the item is pushed into the unconfirmed stack. Once confirmed, it is popped from the stack and pushed into the confirmed stack. Items that have been pushed into the confirmed stack are also popped from the confirmed stack and pushed into the unconfirmed stack if they are the object of addition or correction. Thus, the item that is the current topic of the dialogue is placed on top of the unconfirmed stack. The system then requests confirmation on that item and then on remaining items in the unconfirmed stack. Thus, user-initiated dialogue is conducted while maintaining dialogue consistency.

3.3 Process of Dialogue Management

In the user state, the semantic utterance representation candidates are analyzed based on semantic dialogue constraints. Before evaluating candidates, the dialogue manager deals with ellipses by compensating values (object, size, number) corresponding to the previous system responses. If no appropriate value exists, a default value is given. Fig. 4

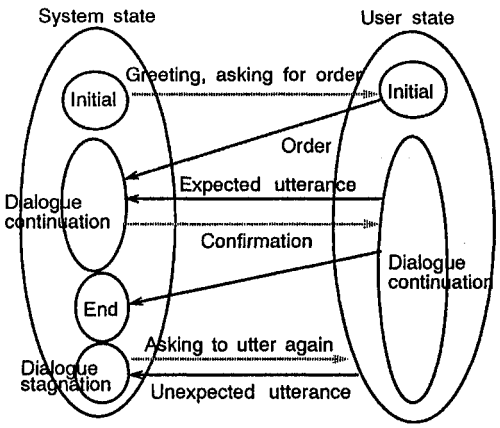

Fig.2 Dialogue model

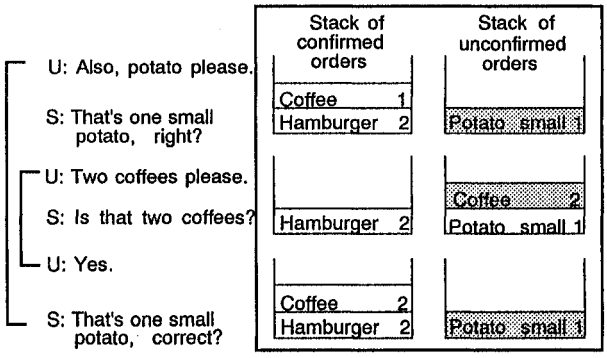

Fig.3 Order table for dialogue topic management

shows the process dealing with a situation where the user answers "No, two please" to the confirmation query "That's one small potato, right?" Here, the syntactic and semantic parser generates two semantic utterance representation candidates. The first candidate has no information on the size and the name of the item; therefore, the item name "potato" and the size "small" are compensated by referring to the previous system response. Next, each of the semantic utterance representation candidates are re-evaluated by checking the consistency of the dialogue history. In the example of Fig. 4, the second candidate shows the deletion of "large cola." Since "large cola" is not in the order table, the score D2 is lowered by the process $D2'=D2*\alpha$ ($\alpha<1$). Because the first candidate is semantically consistent, the score is unchanged ($D1'=D1$). In this way, rescoring is done using heuristic knowledge of the dialogue. After comparing the scores of each candidate, the first candidate is chosen, and a transition to the corresponding system state is made.

In the system state, the dialogue history is modified according to the speech understanding result and then generates semantic response representations. If the user's utterance is not understood, or is not semantically acceptable, the system continues the dialogue by asking the user to repeat the utterance "Please say that once more" and making requests "Please speak up (softly)" based on information from the speech analysis. If the same item is corrected several times, or the request to repeat the utterance is made several times, individual confirmation requests and yes/no

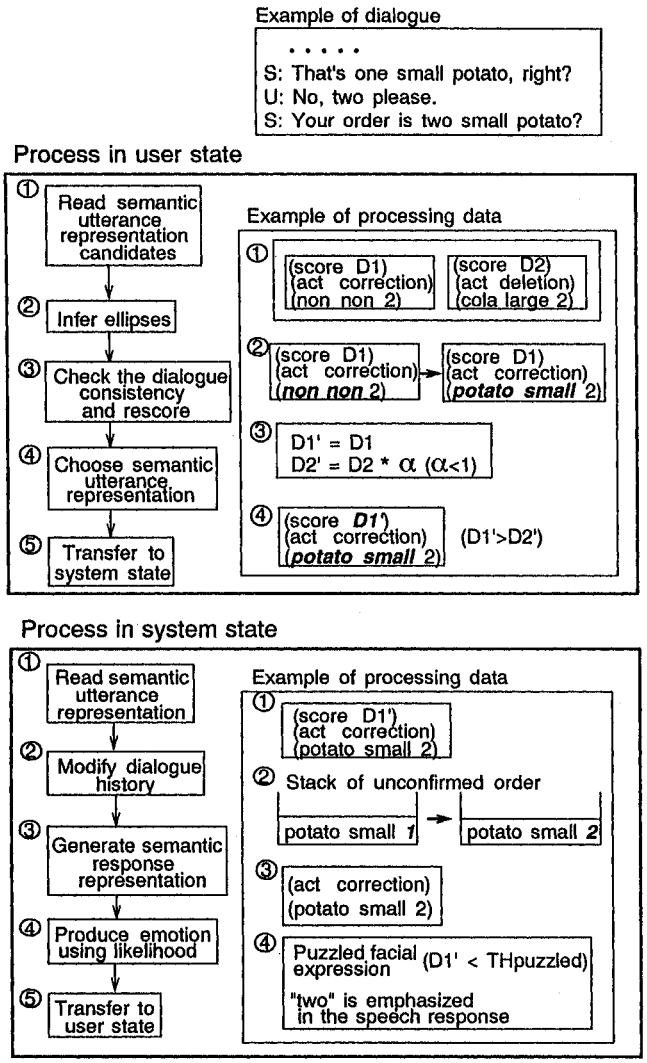

Fig.4 The process of dialogue management

queries are generated to guide the user. The semantic response representation is expressed in a frame form and is added to the likelihood value of semantic utterance representation. This value is used by the response generator to control the intonation of the speech response and the screen display's facial expressions.

4. IMPLEMENTATION

To implement the user-initiated dialogue model, we have represented the dialogue model by nodes and arcs within the ATN (Augmented Transition Network) framework[8]. The system state is comprised of 19 states which deal with taking orders, adding, cancelling, and changing items; the user state is comprised of 8 states and understands the user's utterances according to each of those states. If the user's utterance is not understood, or its semantic utterance representation is unacceptable, the system transfers from a dialogue continuation state to a dialogue stagnation state. The flow of the dialogue is represented in a network of transition between the system states and the user states using ATN. The transition network's grammar is in the following form:

STATE: COMMAND: NEXT: {ACTION}

In the above, STATE is the current state, COMMAND is the command to be executed in the current state, NEXT is the destination state (default state) of the next transition, and ACTION is the action to be taken at the time of transition. It is also possible to make a transition to states other than the default state. COMMAND includes four commands: JUMP, CAT, PUSH, and POP. They are represented in the following form:

(1) JUMP {selection-function}
(2) CAT TOKEN {evaluation-function}
(3) PUSH SUBSTATE {pre-action}
(4) POP

JUMP means an unconditional jump; CAT executes process expressed in TOKEN; PUSH activates the sub-network shown in SUBSTATE; and POP returns the action to the main network from a sub-network. The "selection-function" of the JUMP command changes the dialogue flow relative to the dialogue situation. The "evaluation-function" of the CAT command prunes semantic utterance representation candidates by checking the previous system response and the semantic consistency of the dialogue history.

5. EVALUATION

5.1 Evaluation Environment

We have developed an evaluation environment[9] for TOSBURG II as shown in Fig.5. It consists of a keyword sequence input tool and a dialogue evaluation tool. The keyword sequence input tool is used when a human operator enters a keyword sequence to evaluate the system's speech recognition and understanding. The evaluation requires only the description of user's utterance by a keyword sequence; full transcription is not needed. The dialogue evaluation tool is used to evaluate dialogue speech data and log files TOSBURG II outputs during dialogue. Log files include the final results as well as intermediate processing results, such as keyword lattices, semantic utterance representation candidates, a sequence of dialogue states and the contents of the system's response. The dialogue evaluation tool has a visual interface, shown in Fig. 6, which allows the operator to browse through dialogue data and confirm the meanings of utterances the system inferred from the keyword sequences. Thus, the system performance can be evaluated automatically by comparing the correct sequence with log files.

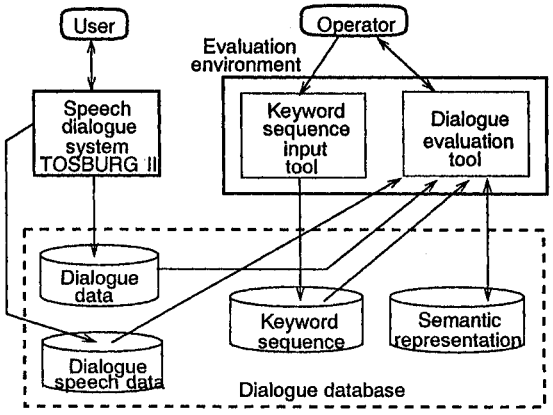

Fig.5 Evaluation environment of TOSBURG II

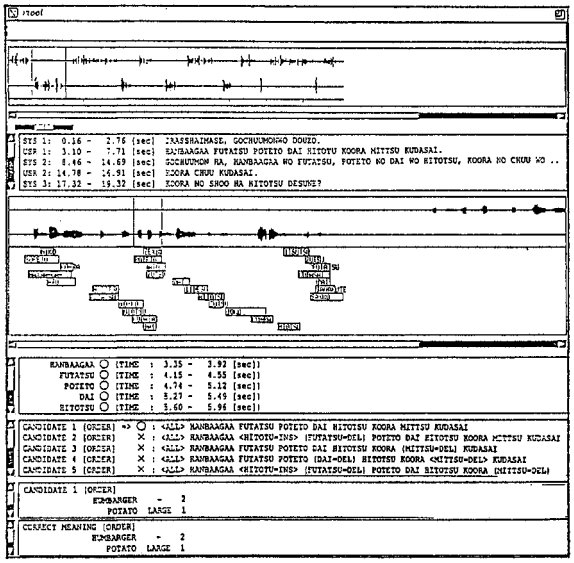

Fig.6 Visual interface of the dialogue evaluation tool

5.2 Experimental Evaluation

We carried out dialogue experiments to evaluate the performance of speech understanding and spontaneous interaction. The experimental conditions and experimental results are given in Tables 1 and 2, respectively. The keyword detection rate is the ratio of correctly spotted keywords to the total keywords; the sentence recognition rate is the ratio of utterance in which all keywords were recognized correctly; the sentence understanding rate is the ratio of correct semantic representations to the total utterances. The latter two rates are about five percent higher under condition 2 than condition 1. In condition 3, all three rates increase significantly compared to the other two conditions because words which are difficult to recognize are not uttered. From these experiments, it is confirmed that restrictions on utterances influence the overall performance of the system.

The performance under condition 1 for each system state is shown in Table 3. In the initial state and in the state of request to repeat utterance, the sentence recognition rate and the sentence understanding rate are below 50%. Because the user tends to order the most of items in the initial state, the speech understanding rate in the initial state are likely to influence the total performance of the system . Therefore, for natural dialogue, we can conclude that it is important to increase the understanding accuracy in the initial state. Specifically, we need to handle utterances that ignored the system's requests, which we term as the user-initiated utterance. After analyzing the log files, we found 91 examples of user-initiated utterances to 420 utterances in dialogue condition 1.

6. CONCLUSION

We have described a user-initiated dialogue model and dialogue processing and presented the evaluation method of TOSBURG II. This system facilitates robust spontaneous human-computer interaction for unspecified users. Experimental results obtained by the real-time system have proven the effectiveness and user-friendliness of the system. It has been confirmed that restrictions on utterances and dialogue conditions influence the overall performance of the system. The ratio of user-initiated utterances is more than 20% when the user utters freely. Therefore, our user-initiated dialogue manager plays an important role in the dialogue progress. It has also been confirmed that the evaluation environment is effective and reduces time and cost for the evaluation of the system's performance.

Table 1 Experimental conditions

Dialogue condition	Subjects /utterances	Explanation
1. Free	38/420	Order several items freely
2. Order specified items	26/280	Hamburger 1, Cheeseburger 2, potato small 1, orangejuice medium 3
3. Free (without size and numer)	20/97	Order several items without size and number

Table 2 Experimental results for each dialogue condition

Dialogue condition	Keyword detection rate (%)	Sentence recognition rate(%)		Sentence understanding rate(%)	
		Top 1	Top 2	Top 1	Top 2
1.	94.8	52.7	56.8	58.2	62.6
2.	94.7	58.2	58.9	63.9	63.9
3.	98.9	88.7	90.7	90.7	96.9

Table 3 Experimental results under condition 1 for each system state

System state	Number of utterances					Total utterances
	Keyword detection	Sentence recognition		Sentence understanding		
		Top 1	Top 2	Top 1	Top 2	
1.	151/155	14	16	17	19	41
2.	204/224	70	76	77	85	121
3.	397/416	124	131	133	142	211
4.	139/146	21	21	22	22	47

System state 1:initial 2:accepting order 3:requesting confirmation 4:requesting to repeat utterance

REFERENCES

[1] Heisterkamp P., McGlashan S. and Youd N. : "Dialogue Semantics for an Oral Dialogue System", *Proc. ICSLP*, Vol.2, Th.fPM.2.3, pp.643-646 (1992).

[2] Allen J. : "Recognizing Intention from Natural Language Utterances", in *Computational Model of Discourse*, M.Brady et al. eds, MIT Press, pp.107-166 (1983).

[3] Grosz B. and Sidner C. : "Attention, Intention, and the Structure of Discourse", *Computational Linguistics*, Vol.12, pp.175-204 (1986).

[4] Takebayashi Y., Nagata Y. and Kanazawa H. : "Noisy Spontaneous Speech Understanding Using Noise Immunity Keyword Spotting with Adaptive Speech Response Cancellation", *Proc. ICASSP*, pp.II-115-II-118 (1993).

[5] Mclaughlin F. : "ICSE 11 Prompts Engineers to Reflect on Yesterday, Look to Tomorrow", *IEEE COMPUTER*, pp.110-112 (1989-07).

[6] Searle J.R. : *Speech Acts*, Cambridge University Press (1969).

[7] Gleaves D. and Nagata Y. : "The Cancellation of Synthetic Speech for a Man-Machine Dialogue System", *Proc. ASJ fall meeting*, 2-5-5 (1991-10).

[8] Woods W.A. : "Transition Network Grammars for Natural Language Analysis", *Proc. CACM*, 13, pp.591-606 (1970).

[9] Seto S., Kanazawa H., Shinchi H. and Takebayashi Y. : "Spontaneous Speech Dialogue System TOSBURG II and Its Evaluation", *Proc. ISSD*, pp.41-44 (1993).