A Study on an E-Mail User Support System Based on Spoken-Language Processing

話し言葉処理に基づく電子メールユーザ支援システムに関する研究

1996年 7月

熊 本 忠 彦

Contents

List of Figures

List of Tables

1. Introduction

In this section, the historical background on studies of dialogue systems is presented, the background and objective of the author's work are described, and the outline and general organization of this article are shown.

1.1 Historical Background on Studies of Dialogue Systems

In the middle of the 1960s, the dialogue system ELIZA was developed by MIT [1]. Although this dialogue system does not analyze the meaning of an input sentence, it does engage in a natural dialogue by using a simple template matching method that extracts keywords from an input sentence, and responds with a template sentence previously prepared as a response for the keywords. ELIZA was of great interest to artificial intelligence researchers due to the smoothness of the dialogue, and marked the beginning of studies on dialogue systems.

In the 1970s, SHRDLU was developed as a dialogue system dealing with the semantics of an input sentence [2]. SHRDLU is a robot simulated on a computer, and works in a given world of blocks where colored blocks of various shapes have been placed and are shown on a display. SHRDLU understands a natural language request, and moves a block in the world of blocks in response to the request. It also responds to queries in a scene of the world of blocks. Wilks has pointed out, however, that most of the faculties of SHRDLU are based on a toy world problem and that the world of blocks is small and restricted, so most of the methods in SHRDLU are not applicable to a larger task domain [3].

The UNIX Consultant (UC) [4] and Put-That-There [5], which are more sophisticated dialogue systems, appeared in the 1980s. The UC is a question answering system that advises users in using the UNIX operating system. This dialogue system is capable of ellipsis resolution and reference disambiguation. The UC, however, does not use situational information such as the state of the UNIX operating system and the progress of the dialogue with a user. In addition, questions can only be input to the UC by typing a sentence on a keyboard; users cannot communicate with the UC through a microphone. Put-That-There is a multimodal system with facilities for speech recognition, simple spoken-language understanding, pointing-gesture recognition, and a large video display. This multimodal system is based on an approach that integrates natural-language references, by using such demonstratives as "this," "it," and "that," and pointing gestures input through a magnetic sensor, so that the ambiguity of referential expressions and rough-and-ready gestures is disambiguated. However, the scope of the spoken-language that the multimodal system understands is very small and restricted in vocabulary and grammar, because the emphasis is on the integration of natural language and pointing gestures.

In the 1990s, many dialogue systems have been developed owing to the rapid progress of hardware technology. These dialogue systems can be roughly classified into four groups.

The focus of the first group, which includes the command interface shell (CIS) [6] and Linta II [7], is on the utilization of the situation. The CIS is a user support system, like the UC, that advises users in using the UNIX operating system. The difference in these two systems is that the CIS automatically checks the preconditions for executing a command, for example, permission to access files and directories, and responds according to the preconditions. Linta II is a human-robot interface implemented into an autonomous mobile robot. Linta II generates situational utterances based on external world information that the robot gets through ultrasonic sensors. Both the CIS and Linta II, however, have limited capabilities for natural language processing.

The second group of dialogue systems includes such multimodal dialogue systems as TOSBURG II [8] and Talkman [9]. TOSBURG II plays the role of a counter-clerk employed in a fast-food restaurant and takes orders from customers. The main features of this dialogue system are that it can understand a customer's speech, carry on a dialogue to take orders, display an animated image of a clerk, respond with synthesized speech and text, and work in real time. But, although TOSBURG II has good potential for use as a total system, it understands only 49 keywords. Talkman is a speech dialogue system that provides facial displays, and it is comprised of a speech dialogue subsystem and a facial animation subsystem. This experimental speech dialogue system was developed to investigate whether the mental barrier that occurs during a dialogue with a machine is reduced by using a facial display. Therefore, as in the TOSBURG II system, the vocabulary and grammar are limited in the Talkman system.

Dialogue systems in the third group have been focused on speech understanding. HARC [10], Phoenix [11], and TARSAN [12] are examples of such speech understanding systems. HARC and Phoenix are used in the ATIS (air travel information service) domain, and they transform input speech into SQL, an artificial language for database retrieval. During the transformation, speech recognition and natural-language understanding modules work in tandem, so that the accuracy of speech understanding is improved. TARSAN is a travel assistant system based on information-retrieval, speech conversation access, and natural-language techniques. It dynamically modifies the size and contents of the speech recognition lexicon for the next utterance according to the result of the information retrieval. This modification process decreases the number of speech recognition errors without increasing the time required for speech recognition. In these three systems, the accuracy of speech understanding is improved by integrating speech recognition and natural-language understanding. However, these systems are still not robust enough to deal with the wide variety of linguistic expressions that appears in spoken language.

The fourth group includes other dialogue systems; for examples, a dialogue system that engages in a creative dialogue with no specified goal [13] and one that allows a user to make timing-free utterances [14]. The former dialogue system is designed for use in situations where users do not have clearly specified goals or change their goals according to a change in viewpoint. In such a situation, a dialogue system must help users clarify or embody their goals by responding with helpful and imaginative answers, and ultimately present satisfactory information to the users. The latter dialogue system allows users to interrupt the synthesized speech uttered by the system, and this timing-free interruption helps realize more natural dialogues. The main focus of these dialogue systems, however, is not on natural-language processing. Therefore, their natural-language processing is very simple.

Although our experimental dialogue system is a user support system, like the UC, it differs from the UC in three ways.

First, the dialogues between a user and the dialogue system are spoken, rather than natural-language dialogues. That is, the user's requests and queries to the dialogue system are given by talking to the dialogue system through a microphone, rather than by typing sentences on a keyboard. However, the speech recognition module of the dialogue system is a manufactured product and is not accurate enough to recognize spontaneous speech. Nevertheless, a spoken-language understanding module, which is one of the primary subjects of our work, has been developed and refined under the assumption that speech recognition accuracy of 100% is attainable. This assumption results from our idea that a speech recognition module with low accuracy for spontaneous speech, which has been a bottleneck in the development of a speech dialogue system, may obstruct the design and development of a spoken-language understanding module.

Second, our dialogue system considers the situation when generating a response to a user's query or request. Since the dialogue system advises users in using software that has a visual interface rather than the UNIX operating system, the relevant situation must be reconstructed from low-level information such as the movement of a mouse and the pressing of a mouse button. In this article, a "situation" is defined as consisting of the state of the target software, the progress of the dialogue, and the event history of the user operations on the target software.

Third, our dialogue system is considered a multimodal system. The input from users now consists of keyboard and mouse operations and speech, while the output to users consists of synthesized speech and text. In the future, input by gesture and eye contact and output by animation of the upper half of a human body will become available in the dialogue system.

1.2 Background of the Author's Work

The recent progress of hardware technology has improved the performance of intelligent systems and increased the number of functions in them. As a result, intelligent systems such as personal computers, word processors, message-recording telephones, and videotape recorders have come to be widely used by ordinary Japanese people. These systems, however, have not been fully utilized because of the complexity of their conceptual structure and the difficulty of their operation. And because their manuals are large and complex, mastering the systems requires a great deal of labor.

The support of an expert consultant is always available for an unskilled user in an ideal environment, but most people are not in such an environment. A user support system that plays the role of an expert consultant is therefore being developed. When a non expert user gets into trouble while performing his task on an intelligent system, he can verbally ask the support system for help. The support system understands the user's speech and responds cooperatively according to both the state of the intelligent system that the user is operating and the progress of dialogue between the user and the support system. Note that the target language here is spoken Japanese.

Such a support system requires at least two kinds of methods: one kind for understanding the user's speech, and the other kind for controlling and managing dialogues with the user. The problems to be considered in developing these methods are described below.

Understanding of User's Speech

For a user support system to understand a user's speech, it needs to have not only a speech recognition technique but also a spoken-language understanding technique adapted to the characteristics of the language [15]. And a user support system that is user-friendly must understand sentences in a user's spontaneous speech, two of the main features of which are the following.

(1) When a speaker makes up a sentence to express a certain communicative intention (abbreviated in this article as CI) to be transmitted to a hearer, the speaker selects — either consciously or unconsciously — one of several expressions that represent the CI equally well.

(2) Such linguistic phenomena as interjections, corrections, hesitations, and inversions may be inserted — either consciously or unconsciously — into the selected sentence.

To understand a spoken sentence, the support system thus needs not only a method for removing or modifying such linguistic phenomena [16]-[18] but also a semantic analysis method that can evaluate the surface structure obtained by removing or modifying them. Studies by Kume et al. and ones by Den et al. are then listed up. Since the design of the semantic analysis method Kume et al. developed [19] is based on general pragmatics, that method seems not to be robust enough to deal with a wide variety of disfluencies which spoken sentences exhibit. Den et al. developed a semantic analysis method [20] whose action is based on the coherence of meanings of the words in a sentence rather than on the syntactic correctness of words. This method, however, will not be put to practical use soon because the ordinary conversation this method is intended to deal with is so flexible that it cannot be constrained by a fixed set of grammatical rules.

Control and Management of Dialogues

A dialogue between a user and a support system is goal-oriented and is conducted so that a user can accomplish his task. The structure of the dialogue thus is complex: a dialogue may have one or more subdialogues, each of which is conducted so that the user can accomplish a subtask. Similarly, a subdialogue may have one or more sub-subdialogues, each conducted so that the user can accomplish a sub-subtask. And the dialogue should be controlled so that the user can accomplish his task effectively.

Recent works on dialogue processing are generally classified into those directed toward understanding a dialogue and those directed toward controlling a dialogue. Works by Iida et al., Takano et al., and Oohira et al. are of the former kind, and works by Nishiyama et al. are of the latter kind. Iida et al. [21] and Takano et al. [22] proposed methods (for understanding a dialogue between people) based on the idea of an "utterance pair." Although an utterance pair is basically a pair consisting of a question and a response, it was extended by each research group to understand more complex dialogue. Oohira et al. [23] proposed a method for managing a dialogue between a user and a dialogue system in database retrieval, but this method is also based on the idea of an "utterance pair." These methods were developed just to understand or manage a dialogue with a particular complex structure, and they therefore do not guide a user to accomplish his task. The "utterance pair" mechanism is thus not suitable for dialogue control.

The target dialogue of Nishiyama et al. [24] is also a dialogue between a user and a dialogue system in database retrieval, but the method they proposed is not for managing the dialogue but for controlling it. Although this method can guide the user to the target data effectively by controlling the responses the system generates, it is not suitable for other task domains because an utterance that should produce a subdialogue is interpreted as change, addition, or deletion of the retrieval condition, and a subdialogue is not produced.

1.3 Objective of the Author's Work

A user support system that helps a user through spoken dialogue is being developed to serve as an intermediary between a naive user and an intelligent system. Dialogues in which the support system can participate are advisory ones and are all related to the use of XMH [25], an X-window-based electronic mail handling program that runs in the X window environment on a UNIX computer and that incorporates, displays, composes, sends, and stores electronic mail. This program has a visual interface and is operated mainly by using a mouse: all the commands are provided as items on pull-down menus and command buttons that are executed by mouse selection or mouse pointing.

When a user gets into trouble while performing tasks, he can ask the support system for help. The support system understands the user's speech and generates a cooperative response appropriate to the state of XMH and the progress of dialogue with the user.

1.4 Outline of the Present Article

As described in subsection **1.2**, at least two kinds of methods should be implemented into the user support system: methods for understanding a spoken sentence, and methods for controlling and managing dialogues. In this article these methods are developed on the basis of analyses of real data [26][27]. Not just any data is suitable for this purpose, and in Section **2** the problems to be considered in collecting this data are discussed and suitable dialogues are collected. This subsection outlines the contents of Sections **2** to **5**.

[Section 2. Design and Construction of an Advisory Dialogue Database]

In this section the design problems associated with constructing a dialogue database are discussed from the viewpoint of collecting dialogues that are useful in developing methods for processing spoken language. XMH usage experiments made to collect such dialogues are shown, and the specifications of an advisory dialogue database constructed from those dialogues are described briefly [28][29].

Advisory dialogues were collected from XMH usage experiments in which an expert consultant provided support to naive XMH users. The user and the consultant were seated in separate sound-isolation booths, and the user was asked to accomplish several specific tasks. The user could use a microphone to ask the consultant for help, and the consultant listened through headphones, generated responses appropriate to the situation, and typed them in Japanese on a computer terminal. The responses appeared on the user's display and were kept there so that the user could refer to them at any time.

The users were 42 undergraduate students (28 women and 14 men) who had almost no computer experience. They therefore first attended a 30-minute lecture on the basic concepts of E-mail, how to operate a mouse and a keyboard, the basic functions of XMH, and how to ask a consultant for help and read the answer. They were also told that they would be able to get information whenever they spoke to the consultant through a microphone.

The experiments were recorded on videotape by a portable camera that also recorded the users' speech. The responses typed by the consultant were saved in text files, and the mouse and key operations were extracted as event information from XMH and were saved in log files. An advisory dialogue database was constructed from these data. The specifications of the dialogue database are these:

Number of user utterances:	855	sentences
Number of consultant responses:	1,397	sentences
Number of mouse operations:	3,949	times
Number of key operations:	34,777	times

And the characteristics of the dialogue database are the following:

(1) The database contains spontaneous advisory dialogues.

(2) The advisory dialogues are all related to the use of XMH.

(3) The database contains not only user utterances and consultant responses but also user operations.

(4) The database consists of text data: when the database was constructed, all user utterances were transcribed into Kanji-Kana mixed sentences. In the XMH usage experiments, all user operations were extracted in text form and saved in log files, and all consultant responses were given as Kanji-Kana mixed sentences that appeared on the user's display.

This kind of dialogue database is a very effective source for developing new methods for processing spoken language and was in fact used to develop a method for understanding a spoken sentence and a method for controlling and managing dialogues with a user.

[Section 3. Understanding of User's Spontaneous Speech]

A subset of user utterance data (475 sentences that 16 users said to a consultant) is obtained from the advisory dialogue database described in Section 2. Of them, the 298 sentences that ten users said are analyzed to determine the types of communicative intentions (CIs) that are observed and the constraints between the surface structure and the CI of a spoken sentence. This section, based on this analysis, defines a form for describing CI and develops a method for constructing a CI description from a spoken sentence. The focus here is on a method for determining the CI type of a spoken sentence [30][31].

A spoken sentence consists generally of two elements: proposition and modality [32]. Proposition is expression which represents objective events or objects or both, and modality is expression which represents a speaker's subjective judgement or attitude. Consider the spoken sentence "ファイルを消したい (I want to delete a file)." This example sentence has elements of the proposition "ファイルを消す (to delete a file)" and the modality "〜したい (want to do)." The CI description of a spoken sentence is thus defined as follows:

$$(\langle \text{CI type} \rangle \ \langle \text{utterance proposition} \rangle),$$

where ⟨utterance proposition⟩ represents a proposition which the spoken-sentence has and is described in frame representation [33]. You find that, in an advisory dialogue, proposition represents just the actions, states, or objects that can be defined in the corresponding task domain. ⟨CI type⟩ represents a modality which the spoken sentence has, and 12 different CI types are defined on the basis of the analysis of the 298 spoken sentences. You find also that, in an advisory dialogue, modality represents just the type of request to a consultant or a user support system. Determining the CI type of a spoken sentence correctly is essential in understanding the spoken sentence. Therefore this article focuses on a method for determining the CI type of a spoken sentence.

Accuracy of the CI determination method was evaluated using the 298 sentences analyzed to develop the method and the 177 sentences not analyzed. The results show that the method was accurate for all 298 of the sentences that were learned data and for 89.3% of the 177 sentences that were unlearned data.

[Section 4. Control and Management of Dialogues with a User]

This section analyzes advisory dialogues that are real data and develops a dialogue processing method adapted to the characteristics of advisory dialogues [34][35]. Dialogue data to be analyzed (corresponding to advisory dialogues between 16 users and a consultant) is obtained from the advisory dialogue database described in Section **2**. This dialogue data has 499 spoken sentences which the users said to the consultant, 652 sentences with which the consultant responded, and 3,495 mouse and 12,796 key operations which the user did.

According to the analysis of the dialogue data, a user-consultant dialogue has some typical dialogue pattern, such as "user-utterance [consultant-instruction, user-operation]." (The notation [A, B] means that an ordered sequence of A and B was found repeatedly.) In addition, the pattern of a dialogue is determined by the CI type of the user utterance with which the dialogue begins. Consequently, a dialogue is controlled by a "process flow" corresponding to the CI type of user utterance with which the dialogue begins. This process flow is a mechanism that is used to control a dialogue and guide the user to his goal, and 12 different process flows are prepared one by one for the 12 CI types defined in Section **3**. The design of the algorithm of a process flow is based not only on the corresponding dialogue pattern but also on the response generation rules [36][37]. Since the design of these rules is based on the know-how of response generation that the human consultant employed in advisory dialogues, the process flows can generate a cooperative response appropriate to the situation.

A user-consultant dialogue may have nested dialogues: subdialogues, sub-subdialogues, and so on. Such nested dialogues did not, however, cross another dialogue. Consequently, process flows for nested dialogues are managed using a stack. The process flow at the top of the stack always runs and controls the nested dialogue. Other process flows do not run while the process flow is running.

The method presented here manages dialogues between a user and a support system by using a stack and controls each of the dialogues by using a process flow so as to guide a user to his task goal.

[Section 5. Construction of an Experimental User Support System]

An experimental user support system based on the spoken-language processing methods developed in Sections 3 and 4 was constructed [38][39]. In this section the architecture of the experimental system is shown and each module of the experimental system is described briefly. The experimental system consists mainly of the Speech Recognition module, Natural Language Processing (NLP) module, Dialogue World model, XMH simulator, Dialogue Processing (DP) module, and Response Generation module. This article describes in detail how the NLP module and the DP module work because the CI recognition method described in Section 3 and the dialogue processing method described in Section 4 were implemented into their modules. Note that the speech recognition system DS200 [40] on the market has been used in the experimental system as the Speech Recognition module. This DS200 transforms continuous speech uttered by an unspecified person into one or more predefined Kanji-Kana mixed sentences.

After three examples of dialogues between a simulated user and the experimental system are shown, how each module of the experimental system works is also shown. In the first example, the experimental system and the user engaged in a typical advisory dialogue. In the second example, they engaged in an advisory dialogue with a subdialogue. And in the third example, the user uttered a sentence fragment to the experimental system in an advisory dialogue.

1.5 General Organization of the Present Article

Section 2 describes the design and construction of an advisory dialogue database. It discusses the design problems associated with constructing a dialogue database, and it describes the construction of an advisory dialogue database based on this discussion. Section 3 describes the understanding of user's spontaneous requests and queries in an advisory dialogue. It focuses on a method for determining the CI type of a sentence spoken by a user. Section 4 describes the control and management of dialogues with a user. The dialogue data obtained from the advisory dialogue database is analyzed, and a dialogue processing method adapted to the characteristics of advisory dialogues is developed. This method manages and controls dialogues with a user so as to guide the user to his task goal. Section 5 describes the architecture of an experimental user support system based on the spoken-language processing methods described in Sections 3 and 4. Section 6 concludes this article and describes the author's plans for future work.

References

[1] Weizenbaum J.: "ELIZA – A Computer program for the study of natural language communication between man and machine," *Communications of the ACM*, Vol. 9, pp. 36-45 (1966).

[2] Winograd, T.: "Understanding natural language," Academic Press (1972).

[3] Wilks, Y.: "Natural language understanding programs within the A.I. paradigm: A survey and some comparisons," in A. Zampolli (ed.), *Linguistic Structures Processing*, North-Holland, Amsterdam, pp. 341-398 (1977).

[4] Wilensky, R., Arens, Y., and Chin , D.: "Talking to UNIX in English: An overview of UC," *Communications of the ACM*, Vol. 27, No. 6, pp. 574-593 (1984).

[5] Bolt, R. A.: "Put-That-There: Voice and gesture at the graphics interface," *Computer Graphics*, Vol. 14, No. 3, pp. 262-270 (1980).

[6] Aoe, J., Maeda, A., Kujime, H., and Morimoto, K.: "Understanding of commands using natural language inputs — An aid to UNIX systems —," *Technical Report of IEICE*, NLC90-44, pp. 25-32 (1990) (in Japanese).

[7] Imai, M., Hiraki, K., and Anzai, Y.: "Human-robot interface with attention," *IEICE Trans.*, Vol. J77-D-II, No. 8, pp. 1447-1456 (1994) (in Japanese).

[8] Takebayashi, Y.: "Spontaneous speech dialogue system TOSBURG II — Towards the user-centered multimodal interface —," *IEICE Trans.*, Vol. J77-D-II, No. 8, pp. 1417-1428 (1994) (in Japanese).

[9] Nagao, K., and Takeuchi, A.: "Speech dialogue with facial displays: Multimodal human-computer conversation," *Proc. 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, pp. 102-109, Morgan Kaufmann Pub., Inc. (1994).

[10] Bates, M., Bobrow, R., Fung, P., Ingria, R., Kubala, F., Makhoul, J., Nguyen, L., Schwartz, R., and Stallard, D.: "The BBN/HARC spoken language understanding system," *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, pp. II-111-II-114 (1993).

[11] Ward, W: "Understanding spontaneous speech: The Phoenix system," *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, pp. 365-367 (1991).

[12] Sakai, K., Yamada, M., Itoh, F., Komori, Y., Ueda, T., and Ikeda, Y.: "Speech guidance system with full text retrieval for texts on CD-ROMs and its evaluation," *IEICE Trans.*, Vol. J77-A, No. 2, pp. 232-240 (1994) (in Japanese).

[13] Takano, A., Hirai, M., and Kitahashi, T.: "Natural language interface on the cooking data base," *Technical Report of JSAI*, SIG-SLUD-9502-2, pp. 8-15 (1995) (in Japanese).

[14] Kikuchi, H., Kudo, I., Kobayashi, T., and Shirai, K.: "Handling of user interruption realizing timing-free utterances for spoken dialogue interface," *IEICE Trans.*, Vol. J77-D-II, No. 8, pp. 1502-1511 (1994) (in Japanese).

[15] Iida, H.: "Pragmatic characteristics of natural spoken dialogues and dialogue processing issues," *Journal of JSAI*, Vol. 3, No. 4, pp. 445-452 (1988) (in Japanese).

[16] Yamamoto, M., Kobayashi, S., and Nakagawa, S.: "An analysis and parsing method of the omission of postposition and inversion on Japanese spoken sentence in dialog," *IPSJ Trans.*, Vol. 33, No. 11, pp. 1322-1330 (1992) (in Japanese).

[17] Sagawa, Y., Ohnishi, N., and Sugie, N.: "Techniques to recover user's utterance failure in a plan-driven dialogue system," *Proc. of Natural Language Processing Pacific Rim Symposium*, pp. 131-138 (1991).

[18] Kanazawa, H., Tsuboi, H., and Takebayashi, Y.: "Key word spotting in continuous speech under noisy environments," *Trans. IEICE*, Vol. J76-D-II, No. 3, pp. 427-435 (1993) (in Japanese).

[19] Kume, M., and Kogure, K.: "Inference mechanism of illocutionary force types for MT systems," *Proc. the 39th Annual Meeting of IPSJ*, 3F-5, pp. 612-613 (1989) (in Japanese).

[20] Den, Y., and Iida, H.: "An information based analysis of conversational sentences," *Technical Report of JSAI*, SIG-F/H/K-9101-12, pp. 111-120 (1991) (in Japanese).

[21] Iida, H., and Arita, H.: "Natural language dialogue understanding on a four-typed plan recognition model," *Trans. IPSJ*, Vol. 31, No. 6, pp. 810-821 (1990) (in Japanese).

[22] Takano, A., Kashioka, H., Hirai, M., and Kitahashi, T.: "A formalization of context of dialogue and a framework for discourse analysis," *Trans. IPSJ*, Vol. 34, No. 1, pp. 88-98 (1993) (in Japanese).

[23] Ohira, E., Abe, M., Komatsu, A., and Ichikawa, A.: "Flexible dialog control on information retrieval," *Trans. IEICE*, Vol. J76-D-II, No. 3, pp. 586-595 (1993) (in Japanese).

[24] Nishiyama, T., and Ooyama, Y.: "Cooperative natural language dialog processing on data base retrieval and its evaluation," *Trans. IEICE*, Vol. J73-D-II, No. 4, pp. 625-632 (1990) (in Japanese).

[25] Peek, J: "Mh and xmh — E-mail for users and programmers," *O'Reilly & Associates, Inc.* (1991).

[26] Kawano, H., Nishio, S., and Han, J.: "Technology of knowledge discovery in databases," *Journal of JSAI*, Vol. 10, No. 1, pp. 38-44 (1995) (in Japanese).

[27] Kumamoto, T., Toraichi, K., Horiuchi, T., Yamamoto, K., and Yamada, H.: "On speeding candidate selection in handprinted Chinese character recognition," *Pattern Recognition*, Vol. 24, No. 8, pp. 793-799 (1991).

[28] Kumamoto, T., Ito, A., and Ebina, T.: "Design and construction of an advisory dialogue database," *IEICE Trans. on Information and Systems*, Vol. E78-D, No. 4, pp. 420-427 (1995).

[29] Kumamoto, T., Ito, A., and Ebina, T.: "Construction of a dialogue database from advisory dialogues between novice computer users and an expert consultant," *Proc. of International Workshop on Sharable Natural Language Resources (SNLR '94)*, pp. 139-146, Nara, Japan (1994).

[30] Kumamoto, T., Ito, A., and Ebina, T.: "Recognizing user communicative intention in a dialogue-based consultant system," *Trans. IEICE*, Vol. J77-D-II, No. 6, pp. 1114-1123 (1994) (in Japanese).

[31] Kumamoto, T., Ito, A., and Ebina, T.: "An analysis of Japanese sentences in spoken dialogue and its application to communicative intention recognition," *Proc. International Conference on Spoken Language Processing (ICSLP '94)*, Vol. 2, No. S17-19.1, pp. 943-946, Yokohama, Japan (1994).

[32] Masuoka, T.: "Modarithi no bunpou (Grammar of modality)," Kurosio Publishers, Tokyo (1991) (in Japanese).

[33] Ito, A., and Kumamoto, T.: "A semantics for a frame representation with circular concept definition," *Trans. IPSJ*, Vol. 34, No. 8, pp. 1731-1740 (1993) (in Japanese).

[34] Kumamoto, T., and Ito, A.: "An analysis of user-consultant dialogue and its application to dialogue control in a dialogue-based consultant system," *Trans. IEICE*, Vol. J77-D-II, No. 8, pp. 1492-1501 (1994) (in Japanese).

[35] Kumamoto, T., and Ito, A.: "An analysis of user-consultant dialogues and its application to dialogue processing in a dialogue-based consultant system," *Systems and Computers in Japan*, Vol. 26, No. 3, pp. 103-114 (1995).

[36] Ito, A., Ebina, T., and Kumamoto, T.: "A taxonomy of the user's questions and response strategy for dialogue based user support," *Trans. IEICE*, Vol. J77-D-II, No. 7, pp. 1319-1328 (1994) (in Japanese).

[37] Ito, A., Ebina, T., and Kumamoto, T.: "Dialogue-based user support for a visual interface software — A case study for XMH —," *Proc. Pacific Rim International Conference on Artificial Intelligence (PRICAI '92)*, 1, pp. 190-196, Seoul, Korea (1992).

[38] Kumamoto, T., Ito, A., and Ebina, T.: "Design and development of a dialogue-based consultant system for supporting the use of electronic mail," *Journal of the Communications Research Lab.*, Vol. 42, No. 2, pp. 161-171 (1995).

[39] Kumamoto, T., Ito, A., and Ebina, T.: "An experimental dialogue-based consultant system for XMH users," *Proc. the 7th Annual Conference of JSAI*, 20-3, pp. 555-558 (1993) (in Japanese).

[40] Osaka Gas Information System Research Institute: "User's manual for speech recognition system DS200," release 1.01 (1992) (in Japanese).

2. Design and Construction of an Advisory Dialogue Database

In this section, the design problems associated with constructing a spoken dialogue database are discussed from the viewpoint of advisory dialogue collection, XMH usage experiments to collect advisory dialogues are described, and the dialogue database constructed from the collected dialogues is shown [1][2].

2.1 Introduction

A two-way dialogue is one of the most effective means of communication between people, and between people and machines. For a machine to engage in a dialogue with its user, it should not only be able to recognize and synthesize speech, but it should also be able to understand and generate spoken language [3]. However, spoken language is difficult to process. Especially, spoken Japanese is highly flexible, and cannot be constrained in a fixed set of grammar rules.

To develop methods for processing spoken Japanese, it is essential to analyze real dialogues and to find the characteristics of spoken Japanese [4]. For this purpose, several dialogue databases have so far been constructed and released to the public [5]-[7]. Based on analyses of these databases, several methods have already been developed for processing spoken language [8][9].

However, the dialogues in these databases were produced by subjects who were forced to engage in experimental dialogues. In such a forced situation, a subject may say only what is necessary, and may put higher priority on the dialogue itself rather than the actual task goals. In real situations, humans usually give priority to achieving their task goals. They ask the consultant for help only when it is necessary for achieving the task goals. They may say unnecessary things, may forget to say necessary things, and may not know even what to ask. Therefore, all the dialogues obtained in a forced situation cannot be regarded as being spontaneous.

A new dialogue database with the following four features was therefore constructed.

(1) The dialogues are advisory ones between naive computer users and an expert consultant. These are goal-oriented and consist basically of questions and answers. In the advisory domain, user support systems are expected to be realized in the near future.

(2) The advisory dialogues are all related to the use of XMH [10], X-window-based electronic mail handling program that has a visual interface operated by a keyboard and a mouse. Visual interface softwares such as XMH are generally easy for naive users to operate, because all the commands are provided as items on pull-down menus and command buttons which can be operated using the mouse. However, simple mistakes often cause fatal errors and can put naive users into deadlock situations because the visual interface softwares do not give them possible recovery methods. In such situations, simple and direct advice is the most useful way to solve the problems of naive users. Such advice can be generated by user support systems.

(3) In the experiments for collecting advisory dialogues, the primary objective of the users was not to engage in dialogues but to achieve specific task goals using XMH. The users could verbally ask a consultant for help when necessary. The consultant helped them solve their problems through dialogues. Although the average number of utterances per user was about 20, one of the users didn't ask for help at all and achieved the task goals through trial and error. This wouldn't have happened in the forced situation mentioned above.

(4) The dialogue elements include not only what the users said, like conventional databases, but also the XMH operations performed by the users. This is because it is important to check

the user operations and manage the XMH states to give effective advice to the users. The dialogue database is useful also for developing new methods for multimodal user support systems because the database consists of two modes, i.e., utterances and XMH operations.

In the next subsection, the design problems associated with constructing a spoken dialogue database are discussed from the viewpoint of advisory dialogue collection. Subsection **2.3** describes how the advisory dialogues were collected and what dialogues were collected. Subsection **2.4** is a conclusion of this section.

2.2 Design of a Dialogue Database

Generally speaking, dialogues can be classified into three types [3]; goal-oriented ones, goal-free ones and mixed ones. Goal-oriented dialogues appear in contexts such as consultation, information retrieval, and inquiry. Since dialogue systems are in great demand in these domains, they are expected to be put to practical use in the near future. Goal-free dialogues are desultory conversations, chats, and so on. For a machine to be able to engage in this type of dialogue, many problems must be solved in both speech processing and spoken-language processing. It would take years before all the problems are solved. Mixed dialogues are, for example, discussions. Automatic processing of these dialogues is more difficult than that of goal-oriented dialogues. Consequently, at present, the target dialogues should be goal-oriented.

When a human communicates with a machine through spontaneous dialogues, he/she should be able to communicate through spontaneous spoken language instead of formal written language. However, spontaneous utterances with no constraints cannot yet be managed by the machines. On the other hand, explicit constraints make dialogue systems unfriendly. A dialogue domain must be therefore selected in which what one says and how one says it are automatically restricted. Taking the above into consideration, advisory dialogues was selected, which consist basically of questions and answers.

The dialogues in the ATR Dialogue Database (ADD) [5] are goal-oriented inquiries. The dialogues in the ASJ continuous speech corpus [6][7] are goal-oriented guidances. These dialogues also consist basically of questions and answers.

Recently, the "Wizard of Oz" setting has become a popular way of collecting goal-oriented dialogues. In this setting, an experimenter and a subject engage in a dialogue, and the experimenter pretends to be a dialogue system. The subject is not informed of the fact, and is made to believe that the other participant is a computer. In such a situation, it is known that the subjects tend to speak to the fake dialogue system using very simple language [11]. If real dialogue systems have insufficient conversational abilities and the users are unfamiliar with the systems, the dialogues between the systems and the users will probably be similar to those collected in the "Wizard of Oz" setting.

However, people are expecting a dialogue system that behaves like a human consultant. Since the users are expected to speak to such a system in just the same manner as they would to a human, the "Wizard of Oz" setting was not adopted. Hence, the subjects in the experiments were told that a human consultant played the support system's role.

The know-how of spontaneous dialogue collection listed in Ref. [7] is as follows.

(1) A person who plays the dialogue system's role should be an expert on the target domain.

(2) It is important that the expert understands exactly the system's performance and limitations, as well as his/her role.

(3) It is helpful if the people acting as users understand what the system can do.

(4) Excessively detailed examples should not be given to the user just before the experiments.

(5) The user should not actually know the answer to his/her question.

(6) The user should not start the experiment without fully understanding his/her role.

(7) Shifts in the initiative of the dialogue from one speaker to the other should be completely flexible.

(8) The environment of the dialogue experiments should be similar to the actual dialogue situation.

(9) Unless there are particular reasons for selecting a task domain, one should select tasks with which the users are familiar in their daily lives.

Here, the above nine items are discussed from the viewpoint of advisory dialogue collection. Items (1), (4), (5), (6), (7), (8) and (9) are important also in collecting advisory dialogues. However, item (2) is object-dependent. When collecting dialogues as basic data to determine the specifications of a dialogue-based user support system, the users should ask what they want to ask without taking the other dialogue participant's response ability into consideration. This is the only way of finding out what the users want to request. Item (3) is desirable, but cannot be relied on in practice, because people who ask the support system for help should not be expected to have mastered how to use it. Item (5) is a natural requirement. The condition is also added that the user truly wants to know the answer to the query. This is the necessary condition for spontaneous dialogue collection. If the primary objective of the users is not to engage in a dialogue but to achieve specific task goals, this condition is satisfied automatically. Item (6) is almost right. However, it is not easy for a user to play his/her role perfectly. If the user puts higher priority on the dialogue itself rather than the actual task goals, he/she may say only what are necessary. In real situations, the user may say unnecessary things, may forget to say necessary things, and may not know even what to ask. These are inevitable features of spontaneous dialogues. Hence, dialogues are collected under the situation that a user puts higher priority on achievement of the task goals.

A dialogue database was constructed based on the above considerations and discussion.

2.3 Construction of a Dialogue Database

2.3.1 Collection of advisory dialogues

Advisory dialogues were collected from XMH usage experiments in which an expert consultant provided support to naive XMH users. The setup of the experiments is shown in Fig. 2.1. The two dialogue participants — the user and the consultant — were each seated in sound-isolation booths. Using XMH, the user was required to achieve several specific task goals: making arrangements for a tea party and so on. Whenever necessary, the user could verbally ask the consultant for help through a microphone. The consultant listened to the user through headphones, made responses appropriate to the situation, and typed them in Japanese on a computer terminal. The responses were given not in voice, but in text form which appeared on the user's display. The text was kept on the user's display so that the user could refer to it any time. Figure 2.2 shows the operation window of XMH with a help window. In Fig. 2.2, the left window is a help window displaying consultant responses. The right window is an XMH window. The three messages in the XMH window give instructions for the user's task goals.

The specifications of the experiments are shown below. The users were 42 undergraduate students (28 women and 14 men) who had practically no computer experience. They therefore attended a 30-minute lecture on the following subjects: the basic concepts of E-mail, how to operate a mouse and a keyboard, the basic functions of XMH, and how to ask a consultant

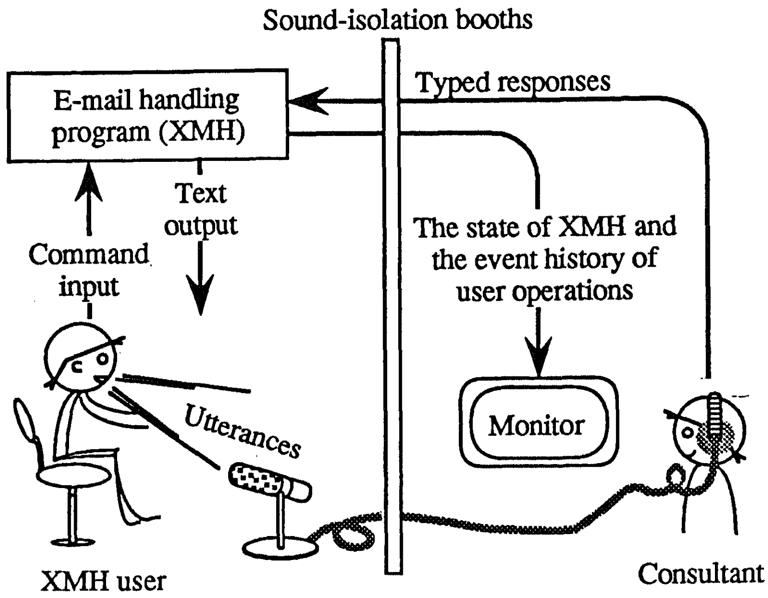


Figure 2.1 XMH usage experiment setup.

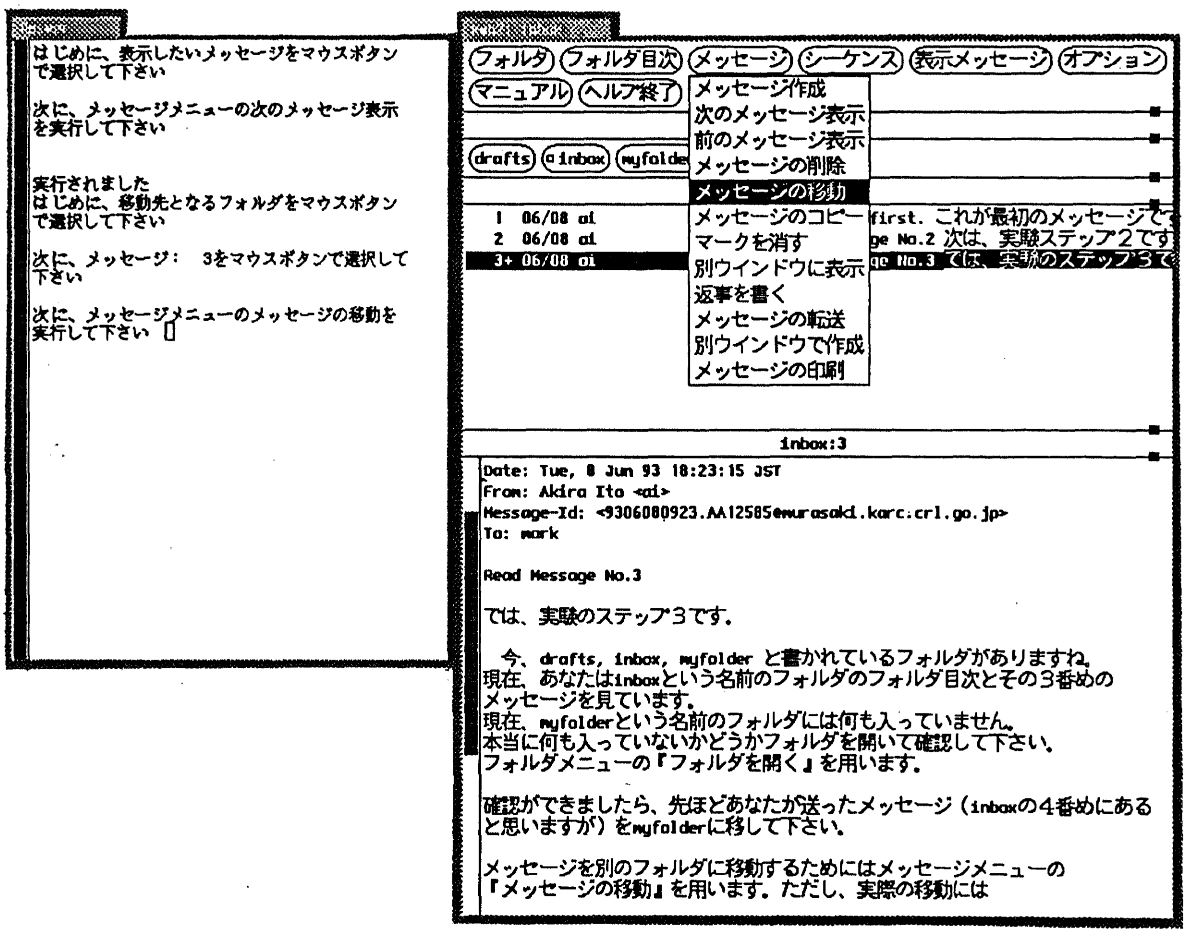


Figure 2.2 Operation window of XMH with a help window.

for help and read the answer. The users were told that they would be able to get information whenever they spoke to the consultant through a microphone.

The experiments were recorded on videotape by a portable camera, which also recorded the users' utterances. The responses typed by the consultant were saved in text files, and the mouse and key operations were extracted as event information from XMH, and saved in log files.

2.3.2 Transcript of user utterances in the dialogues

The user utterances were transcribed from the videotapes, and only transcription errors (omitted characters and wrong characters) were carefully removed. Linguistic phenomena such as interjections, repairs, hesitations, and paraphrases and short sentence fragments were transcribed exactly. Sentence units were extracted using the method of Ref. [5], in which a chain of semantically connectable speech is regarded as a sentence. If necessary, grammatical representations (postpositional words representing conjunctions and the ends of sentences) and prosodic information (pauses and intonation) were used to determine the ends of sentences.

The dialogue data is a direct transcription of what the users said. The dialogues shown in Ref. [5] were subjected to preliminary analyses to extract information such as the correspondences between Japanese and English words, which were attached to the transcribed dialogues. Several methods have been developed based on these preliminary analyses [12][13]. However, the preliminary analyses might limit the utilization of the transcribed dialogues since they themselves are problems that must be solved. Hence, any preliminary analysis was not done. Note that the transcribed utterances involve punctuation marks "、" and "。", pause mark "…" and question mark "？" as speech information.

Two sample dialogue transcripts are shown below; the symbols "U" and "C" denote the user utterance/operation and the consultant response, respectively. The numbers following these symbols are serial numbers. To make it easier to follow the transcripts, the user operations are shown in square brackets. The parenthesized sentences are the English versions of the utterances.

Dialogue No. 1

U1: どうやって移動するんですか？
(How can I move a message?)

C2: はじめに移動するメッセージを選択して下さい
(First, select the message that you want to move.)

U3: [Selects message "4."]

C4: つぎに、移動先のフォルダを選択して下さい
(Next, select the destination folder.)

U5: [Selects folder "myfolder."]

C6: それから、メッセージメニューの中のメッセージの移動、を実行して下さい
(Next, execute the "MoveMessage" command on the "Message" menu.)

U7: [Executes "MoveMessage" command.]

C8: 最後に、フォルダ目次の中の変更の実行、を実行して下さい
(Lastly, execute the "CommitChanges" command on the "TableOfContents" menu.)

U9: [Executes "CommitChanges" command.]

C10: 移動されました
(The message was moved.)

Dialogue No. 2

U11: 手紙の内容が出てこないんですけど。
(The contents of the message are not displayed.)

C12: メッセージ2を選択して下さい。
(Select message 2.)

U13: [Selects message 2.]

C14: つぎに、メッセージメニューの中の次のメッセージを実行して下さい
(Next, execute the "NextMessage" command on the "Message" menu.)

U15: [Wrongly executes "ComposeMessage" command, which is next to "NextMessage" command.]

U16: 二つ画面が出てきたんですけど。
(Another window appeared.)

C17: 新しい画面の左下にある終了、を押して下さい。
(Push the "Close Window" button at the bottom left-hand corner of the new window.)

U18: [Pushes "Close Window" button.]

U19: [Executes "NextMessage" command.]

2.3.3 Specifications of the dialogue database

A dialogue database was constructed, which consists of 42 data sets. Each data set consists of four files. The first one contains the transcripts of the spoken dialogues between each user and a consultant. The second and third files contain descriptions of the mouse operations and key operations, respectively. The last one contains basic statistical data of the dialogues.

First, the data format of the file containing the transcribed dialogues is shown.

An utterance is described in the following form.

(*U-time Speaker Utterance*)

U-time shows the time when a user uttered, and was counted up in seconds since XMH was started up. *Speaker* is either "user" or "consultant." Although the user actually spoke to the consultant, the consultant only typed the responses in Japanese. *Utterance* is the sentence or short sentence fragment which the user said, or the sentence or part of sentence which the consultant typed in. When *Speaker* is "user", *Utterance* involves punctuation marks "、" and "。", pause mark "…" and question mark "?" derived from the speech information. When *Speaker* is "consultant," *Utterance* is just one line which the consultant typed. Since the consultant tried to make a quick response, a response was often divided into more than one line. This enabled the consultant to start responding within a few seconds.

Examples 1:

(2204 user "どうやって移動させたらいいのかわからないんですけど。")
This *Utterance* corresponds to the English sentence "I don't know how to move it."

(2207 consultant "はじめに、")
(2216 consultant "移動させたいメッセージを選択して")
(2218 consultant "下さい")
These three *Utterances* correspond to the English sentence "First, select the message that you want to move."

Next, the data format of the two files containing descriptions of the XMH operations is shown. A key operation is described in the following form.

(*O-time* KEY *Place* *Key*)

O-time shows the time when a user did a key operation or a mouse operation, and was counted up in milliseconds since XMH was started up. *Place* indicates the cursor position when the user typed the key. *Key* shows what key was typed. For example, a right cursor-move key and a delete key are described in the symbols of "RIGHT" and "DEL", respectively. All the *Places* and *Keys* observed in the experiments are shown in Table 2.1.

Examples 2:

(1193.21 KEY COMP A)

Table 2.1 All the *Places* and *Keys* observed in the experiments.

(a) *Keys* observed when *Place* is "COMP."

0 2 3 4 5 6 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
+ − / [ ] ˆ _ } ˜ = ? @
BACK-QUOTE BS CNTL COLON COMMA CR CTRL-A DEL DOUBLE-QUOTE
DOWN ESC KAKKO-BEGIN KAKKO-END LEFT NIL PERIOD QUOTE RIGHT
SEMI-COLON SHARP SHIFT SPACE TAB UP

(b) *Keys* observed when *Place* is "HELP."

SHIFT

(c) *Keys* observed when *Place* is "TOC."

A J M N O R T
SHIFT SPACE UP

(d) *Keys* observed when *Place* is "VIEW."

9
A G I J M N O R T Y
/
BS CR DOWN LEFT NIL RIGHT SPACE TAB UP

(e) *Keys* observed when *Place* is "XMH2."

0 1 2 3 4 5 6
A B C D E F G H I J K L M N O P Q R S T U W X Y Z
! + − / < > ? @ [ ] _ ˜
BACK-QUOTE BS CNTL COLON COMMA CR CTRL-A DEL DOUBLE-QUOTE
DOWN KAKKO-BEGIN KAKKO-END LEFT NIL PERIOD QUOTE RIGHT
SEMI-COLON SHIFT SPACE UP

(1194.498 KEY COMP RIGHT)
(1195.068 KEY COMP RIGHT)

A mouse operation is described in the following form.

(*O-time* MOUSE *Menu* *Command* *Argument*)

Menu and *Command* identify which menu was opened and which command on the menu was executed, respectively. *Argument* is the argument of the command, if it exists. All the *Menus*, *Commands* and *Arguments* observed in the experiments are shown in Table 2.2.

Examples 3:

(3232.116 MOUSE NIL SELECTMESSAGE (5))
(3252.818 MOUSE NIL SELECTFOLDER MYFOLDER)
(3262.495 MOUSE MESSAGE MOVE)
These three lines represent that message 5 was selected, that folder "myfolder" was selected, and that the "MoveMessage" command on the "Message" menu was executed, respectively.

Table 2.2 All the *Menus*, *Commands* and *Arguments* observed in the experiments.

(a) When a command on a pull-down menu is executed.

Menus	*Commands*
FOLDER	CLOSEWINDOW CREATEFOLDER OPENFOLDER OPENFOLDERINNEWWINDOW
MANUAL	CLOSEWINDOW COMMITCHANGES COMPOSE COPY DELETE DELETEFOLDER EDITMESSAGE FORWARD HELP INCORPORATENEWMAIL MOVE OPENFOLDER PACKFOLDER REPLY RESCANFOLDER SORTFOLDER UNMARK VIEWINNEW VIEWNEXTMESSAGE
MESSAGE	COMPOSEMESSAGE COPY DELETE FORWARD MOVE REPLY UNMARK USEASCOMPOSITION VIEWINNEW VIEWNEXTMESSAGE VIEWPREVIOUS

(b) When a command button is pushed.

Commands	*Arguments*
COMPWINDOW_BUTTON	CLOSE COMPOSE RESET SAVE SEND
DIALOG_BUTTON	CANCEL CONFIRM NO OK VALUE YES
HELP_BUTTON	QUIT
MAN_BUTTON	EXIT VALUE
SELECTFOLDER	DRAFTS INBOX MYFOLDER
SELECTMESSAGE	(a list of numbers) or NIL
VIEW_BUTTON	CLOSE
TABLEOFCONTENTS	COMMITCHANGES INCORPORATENEWMAIL PACKFOLDER SORTFOLDER

Table 2.3 Statistical data of the database.

User No. (Gen.)	# of Ut.	# of Re.	# of Mouse Op.	# of Key Op.	User No. (Gen.)	# of Ut.	# of Re.	# of Mouse Op.	# of Key Op.
1 (F)	49	83	100	1,113	22 (F)	7	14	78	1,117
2 (F)	17	28	89	1,004	23 (M)	1	7	70	892
3 (F)	41	76	64	740	24 (M)	1	6	148	522
4 (F)	7	15	87	713	25 (F)	45	83	113	799
5 (F)	15	39	74	817	26 (F)	23	65	75	511
6 (F)	4	14	165	697	27 (F)	40	52	94	111
7 (M)	9	17	100	877	28 (M)	15	28	113	1,223
8 (M)	2	14	108	806	29 (M)	73	91	71	936
9 (M)	3	8	64	1,135	30 (M)	17	32	53	380
10 (F)	93	109	51	683	31 (F)	20	35	74	320
11 (F)	9	12	118	618	32 (F)	57	86	110	990
12 (M)	5	20	160	750	33 (F)	54	62	83	899
13 (M)	1	5	174	1,720	34 (F)	7	12	74	452
14 (F)	9	12	101	752	35 (F)	67	70	132	899
15 (M)	6	20	91	664	36 (F)	33	41	43	591
16 (M)	0	8	173	1,059	37 (F)	12	21	59	493
17 (F)	22	32	88	783	38 (F)	17	27	66	857
18 (F)	3	8	84	490	39 (F)	15	23	95	956
19 (M)	7	18	76	1,299	40 (F)	4	9	177	709
20 (M)	1	5	52	795	41 (F)	17	26	68	1,301
21 (F)	9	25	61	677	42 (F)	18	39	73	1,627
					Total	855	1,397	3,949	34,777

Gen.: gender, F: female, M: male,
Ut.: user utterances, Re.: consultant responses,
Mouse Op.: mouse operations, Key Op.: key operations.

The "MoveMessage" command moves a message from the current folder to another folder and requires a target message and a destination folder. In Examples 3, the command was executed successfully.

Lastly, the contents of the file containing basic statistical data of the dialogues are described. This file contains the number of user utterances, the number of consultant responses, the number of mouse operations and the number of key operations. Table 2.3 shows the statistical data for 42 data sets, and gender of the users.

2.4 Conclusion

The target of this research is to develop a user support system which helps naive computer users solve their problems through spoken dialogue [14][15]. For that purpose, real dialogues must be analyzed and the characteristics of spoken Japanese must be found.

Advisory dialogues were collected from the XMH usage experiments and a dialogue database was constructed. The main features of the database are as follows:

(1) The target dialogues were spontaneous advisory ones.

(2) The advisory dialogues were all related to the use of XMH that has a visual interface operated by a keyboard and a mouse.

(3) Not only user utterances and consultant responses but also XMH operations are included as dialogue elements.

(4) The dialogue database consists of text data; All user utterances were transcribed in constructing the dialogue database. In the XMH usage experiments, all XMH operations were saved in text form, and all consultant responses were given as Kanji-Kana mixed sentences which appeared on the user's display.

This kind of dialogue database is a very effective source for developing new methods for processing spoken Japanese in dialogue-based user support systems. In fact, based on analyses of the advisory dialogue database, a method for understanding a user utterance and a method for controlling and managing a dialogue with a user are developed in Sections **3** and **4**, respectively. Note that the dialogue database is available to people only for academic uses.

Speech data also are useful in developing new methods for processing speech. In the XMH usage experiments, the user speech was recorded on the videotapes. Currently, we are considering specifications of a speech database to be constructed.

References

[1] Kumamoto, T., Ito, A., and Ebina, T.: "Design and construction of an advisory dialogue database," *IEICE Trans. on Information and Systems*, Vol. E78-D, No. 4, pp. 420-427 (1995).

[2] Kumamoto, T., Ito, A., and Ebina, T.: "Construction of a dialogue database from advisory dialogues between novice computer users and an expert consultant," *Proc. of International Workshop on Sharable Natural Language Resources (SNLR '94)*, pp. 139-146, Nara, Japan (1994).

[3] Iida, H.: "Pragmatic characteristics of natural spoken dialogues and dialogue processing issues," *Journal of JSAI*, Vol. 3, No. 4, pp. 445-452 (1988) (in Japanese).

[4] Shirai, K., and Takezawa, T.: "Spoken dialogue processing," *Journal of JSAI*, Vol. 9, No. 1, pp. 48-56 (1994) (in Japanese).

[5] Ehara, T., Ogura, K., Shinozaki, N., Morimoto, T., and Kurematsu, A.: "Construction of a dialogue database (ADD) from dialogues through telephone or keyboard channels," *Trans. IPSJ*, Vol. 33, No. 4, pp. 448-456 (1992) (in Japanese).

[6] Itahashi, S.: "Creating speech corpora for speech science and technology," *IEICE Trans.*, Vol. E74, No. 7, pp. 1906-1910 (1991).

[7] Kobayashi, T., Itahashi, S., Hayamizu, S., and Takezawa, T.: "ASJ continuous speech corpus for research," *J. Acous. Soc. Japan*, Vol. 48, No. 12, pp. 888-893 (1992) (in Japanese).

[8] Yamamoto, M., Kobayashi, S., and Nakagawa, S.: "An analysis and parsing method of the omission of postposition and inversion on Japanese spoken sentence in dialog," *Trans. IPSJ*, Vol. 33, No. 11, pp. 1322-1330 (1992) (in Japanese).

[9] Kudo, I., and Tomokiyo, M.: "An ellipsis-resolution mechanism by using Japanese predicate particularity," *Trans. IEICE*, Vol. J76-D-II, No. 3, pp. 624-635 (1993) (in Japanese).

[10] Peek, J.: "Mh and xmh – E-mail for users and programmers –," O'Reilly & Associates, Inc. (1991).

[11] Guindon, R.: "How to interface to advisory system? — Users request help with a very simple language —," *ACM Proc. CHI '88*, pp. 191-196 (1988).

[12] Furuse, O., Sumita, E., and Iida, H.: "Transfer-driven machine translation utilizing empirical knowledge," *Trans. IPSJ*, Vol. 35, No. 3, pp. 414-425 (1994) (in Japanese).

[13] Kudo, I.: "Local cohesive knowledge for a dialogue-machine translation system," *Proc. COLING '90*, Vol. 3, pp. 391-393 (1990).

[14] Kumamoto, T., Ito, A., and Ebina, T.: "Design and development of a dialogue-based consultant system for supporting the use of electronic mail," *Journal of the CRL*, Vol. 42, No. 2, pp. 161-171 (1995).

[15] Kumamoto, T., Ito, A., and Ebina, T.: "An experimental dialogue-based consultant system for XMH users," *Proc. the 7th Annual Conference of JSAI*, 20-3, pp. 555-558 (1993) (in Japanese).

3. Understanding of User's Spontaneous Speech

In this section, a method is developed to recognize user's communicative intentions (CIs) from spoken sentences. Especially, a method for determining the CI type of a spoken sentence is focused on [1][2].

3.1 Introduction

User-friendly dialogue systems must understand a user's spontaneous speech. To achieve this ability, not only a speech recognition technique but also a spoken-language understanding technique adapted to the characteristics of spoken Japanese are essential [3][4]. Den et al. [4] are trying to develop a semantic analysis method for spoken Japanese, but this method may not be put to practical use soon because ordinary conversation is high flexible and cannot be constrained by a fixed set of grammatical rules.

Guindon found that help in an advisory environment is requested in restricted language [5]. Although the target users in his research were English speakers, Japanese speakers also are expected to request help by using language with any constraints.

The approach of this research is based on an analysis of real utterance data. The utterance data are obtained from the advisory dialogue database described in Section **2**, and consist of 475 spoken sentences. 298 of the sentences are analyzed, and constraints between the surface structure of a spoken sentence and communicative intention (CI) of the sentence are investigated. This section, based on the analysis and investigation, defines a description form of user's CI and develops a method for constructing a CI description from a spoken sentence.

A CI description consists of a CI type and an utterance proposition. An utterance proposition represents the "objects," "actions," or "states" that can be defined in a task domain and is described in frame representation. A CI type represents the type of user's attitude to the utterance proposition, and 12 different CI types are defined. Especially, in this article, a method for determining a CI type is focused on.

This section is organized as follows. Subsection **3.2** describes utterance data obtained from the advisory dialogue database, which consist of 475 spoken sentences divided into 16 data sets: 298 sentences in 10 data sets and 177 sentences in 6 data sets. The former sentences are analyzed, and the latter sentences are used to evaluate a CI type determination method to be proposed in this section. Subsections **3.3** and **3.4** analyze the 298 sentences, define a description form of CI, and develop a method for constructing a CI description from a spoken sentence. Especially, a method for determining the CI type of a spoken sentence is focused on. Subsection **3.5** uses the 177 sentences to evaluate performance of the CI type determination method. Subsection **3.6** discusses characteristics of the CI description and CI recognition method. Subsection **3.7** concludes this section and describes our plans for future work in spoken-language understanding.

3.2 User's Spoken Japanese Sentences in the Advisory Dialogue Database

Utterance data was obtained from the advisory dialogue database described in Section **2** and comprised 499 spoken sentences transcribed from what 16 users said to the consultant. Specifications of the utterance data are shown:

Number of spoken sentences:	499	
Total number of characters:	8,147	(16.3 characters/sentence)
Number of data sets:	16	

(Each data set corresponds to a user)

Table 3.1 Construction of the utterance data.

	Data sets for analysis										
No.	1	2	3	4	5	6	7	8	9	10	Total
# of sentences	20	64	59	33	6	46	9	20	7	34	298

	Data sets for evaluation						
No.	11	12	13	14	15	16	Total
# of sentences	73	18	37	21	17	11	177

Table 3.2 Examples of CI types and sentences in the utterance data.

Example sentences	CI types
何を書いたらいいんですか (What should I write here?)	ask-wh:OK
住所忘れました (I forgot his address.)	ask-wh
終了を押せばいいんですね (Should I push the “Close window” button?)	ask-if:OK
メッセージはこれだけでいいんですか (Is this enough for the message?)	ask-if:EQ
これ押さなくちゃ駄目なんですか (Must I push this?)	ask-if:NO
これで移動できてるんですか (Has it been moved successfully?)	ask-if
移すのはどうするんですか (How do I move it?)	ask-how
移動できてないみたいなんですけど (It seems not to have been moved.)	have-belief
スクロールって何ですか (What is “Scroll”?)	ask-about
ロイの次にカーソルを持っていきたい (I want to move the cursor at the next of the letters “roy.”)	have-goal
すみません (Excuse me.)	d-start
わかりました (I got it.)	d-end

The data was 499 sentences consisting of 232 simple sentences, 243 complex sentences, and 24 sentence fragments. Because sentence fragments should be interpreted according to dialogue context [6], they were not the target for the CI type determination method. The 24 sentence fragments were therefore removed from the data.

The 475 simple or complex sentences were divided into two groups: 298 sentences of 10 data sets and 177 sentences of 6 data sets. The former was used for analysis and the latter was used for evaluation of the CI type determination method. Table 3.1 shows construction of the utterance data, and Table 3.2 lists example sentences selected from the utterance data, together with their CI types.

3.3 Description Form of User's Communicative Intention (CI)

A spoken sentence has propositional and modality information [7]. A description for representing the former is called an "utterance proposition" and is described in frame representation [8][9]. A description for representing the latter is called a "CI type" and is selected out of the 12 different symbols to be defined in **3.3.1**. Nuances of a sentence, a user's emotion, and so on can be also expressed as modality information. In user support systems, however, only the CI type of modality information is used because other kinds of information (nuance, emotion, etc.) are not yet technically available. Consequently, the CI of a spoken sentence is described in the following form.

$$(\langle \text{CI type} \rangle \ \langle \text{utterance proposition} \rangle)$$

3.3.1 Classification of CI types

Here the 298 sentences for analysis are classified into 12 groups by three stages, and a different symbol is assigned to each group.

Table 3.3 shows the first two stages by which the sentences are classified into eight groups and the 8 different CI types assigned to each group. Table 3.4 shows the number of the sentences in each group. In the first stage the 298 sentences were classified on the basis of the sentence type [10], and in the second stage they were classified from the viewpoint of what the users required of the consultant.

A CI description is used in our experimental user support system [11][12] under development. Especially, the CI type works to determine a response procedure for the corresponding utterance proposition. Hence, the questions belonging to CI types "ask-if" or "ask-wh" should be subclassified because response procedures differ between the questions on future events, questions on past actions or states, and questions on the others. The first questions concern a task plan and responding to them requires task plan knowledge. The second questions concern events in XMH and responding to them requires information in the database on events in XMH. This database has been implemented in the support system as the Object World model [9]. The

Table 3.3 CI types and their classification criteria.

The first stage			The second stage	CI types
Differentiated sentences	Interrogative sentences (Questions)	Wh questions	To get instructions	ask-how
			To get attribute values	ask-wh
			To get metaknowledge	ask-about
		Yes/No questions	To get truth values	ask-if
	Declarative sentences		To express a task goal	have-goal
			Otherwise	have-belief
	Imperative sentences	(There were no imperative sentences in the utterance data.)		
	Exclamatory sentences	(These are dealt with as declarative sentences.)		
Undifferentiated sentences			To start a dialogue	d-start
			To end a dialogue	d-end

Table 3.4 The number of sentences classified into each CI type.

CI types	# of sentences	CI types	# of sentences
ask-how	82	have-goal	4
ask-wh	27	have-belief	20
ask-about	4	d-start	28
ask-if	128	d-end	5

Table 3.5 Subcategorization of CI types "ask-if" and "ask-wh."

Question contents	ask-if	ask-wh	Sub CI types
Permission of future event			
(by positive expressions)	52	8	OK
(by negative expressions)	3	0	NO
Event in XMH	54	19	nil (omissible)
Otherwise	19	0	EQ
Total	128	27	

last questions concern a task domain and responding to them requires task domain knowledge. Moreover, the questions on a task plan should be divided into positive ones and negative ones. The former are the questions in which permission for an event is applied for by the method of asking about the acceptability of its event. For example, this kind of questions have such a surface structure as "～したらいいんですか (Should I ～?)." Note that these questions in Japanese correspond to the English sentences beginning with "Should I." The latter are the questions in which permission for an event is applied for in the form of prohibition. For example, this kind of questions have such a surface structure as "～しなければいけないんですか (Must I ～?)." Note that these questions in Japanese correspond to the English sentences beginning with "Must I." Table 3.5 lists the subtypes which were assigned to the subclassified sentences. The numbers in Table 3.5 are those of the sentences into which the 128 sentences with "ask-if" and the 27 ones with "ask-wh" were subclassified.

Classification of CI types is described in the extended Backus form.

⟨CI type⟩ ::= ask-how | ask-about | ⟨ask-wh⟩ | ⟨ask-if⟩ | have-goal | have-belief
| d-start | d-end
⟨ask-wh⟩ ::= ask-wh | ask-wh:OK
⟨ask-if⟩ ::= ask-if | ask-if:OK | ask-if:NO | ask-if:EQ

3.3.2 Description of an utterance proposition in frame representation

An utterance proposition represents the "objects," "actions," or "states" that can be defined in the task domain and is described in frame representation.

Frames are classified into object frames and event frames. Object frames represent the objects that can exist in the task domain and event frames represent the events that can occur in the task domain. Event frames are further classified into action frames representing acts and state frames representing states. Examples of frames are shown in Fig. 3.1.

A frame has a name, a number, and a set of slots. The frame name represents the class of

#<PushCommandButton #X139EG76>
target slot value is #<CommandButton #X139EB8E>

(a) An action frame representing the act of pushing the 終了(Close window) button.

#<CommandButton #X139EB8E>
name slot value is "終了(Close window)"

(b) An object frame representing the 終了(Close window) button.

Figure 3.1 Frames representing an action and an object.

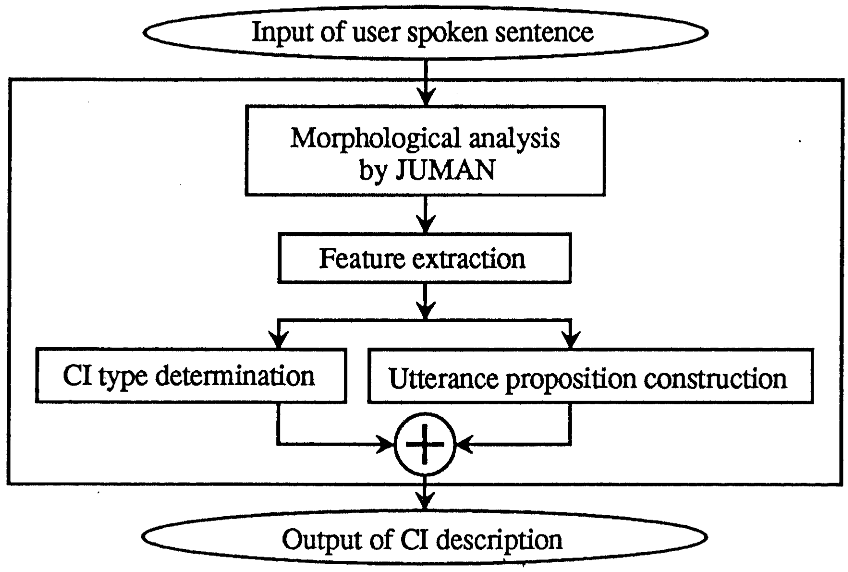


Figure 3.2 Flow of communicative intention recognition.

the object or event that the frame represents. The frame number is an ID number to distinguish different objects or events in the same class. And the slot has a name and a value; it represents an attribute of the object or event which its frame represents.

For example, the act of pushing the "Close window" button is described as the action frame shown in Fig. 3.1(a) and the "Close window" button is described as the object frame shown in Fig. 3.1(b). This object frame has been bound in the target slot of the action frame.

Construction of an utterance proposition is also described in the extended Backus form.

$$\langle\text{utterance proposition}\rangle ::= \langle\text{proposition}\rangle$$
$$\langle\text{proposition}\rangle ::= \text{frame} \mid \langle\text{and-form}\rangle \mid \langle\text{or-form}\rangle$$
$$\langle\text{and-form}\rangle ::= (\text{and } \langle\text{proposition}\rangle\ \langle\text{proposition}\rangle^*)$$
$$\langle\text{or-form}\rangle ::= (\text{or } \langle\text{proposition}\rangle\ \langle\text{proposition}\rangle^*)$$

Table 3.6 Example I/O in the morphological analysis stage.

Input sentence: 終了を押せばいいんですね
(Should I push the Close window button?)

Output:

Morphemes	Original forms	Parts of speech		Inflection forms
終了	終了 (Close window)	noun	action noun	*
を	を ("wo")	postpositional word	postpositional word representing case	*
押せば	押す (push)	verb	*	basic conditional form
いい	いい (OK)	adjective	*	basic form
んです ("Ndesu")	んだ ("Nda")	auxiliary verb	*	"desu"-type of basic form
ね	ね ("ne")	postpositional word	postpositional word indicating end of sentence	*

Table 3.7 Example I/O in the feature extraction stage.

Feature values for CI type determination	Morphemes	Feature values for utterance proposition construction	
action noun	終了	終了 (Close window)	action noun
	を	を ("wo")	postpositional word representing case
action_predicate basic conditional form	押せば	押す (push)	verb
OK_predicate	いい		
	んです		
pp_word_interrogation	ね		
end_of_sentence			

3.4 CI Recognition Method

The CI recognition flow is shown in Fig. 3.2, and each stage of the flow is described below.

⟨Morphological analysis stage⟩

In this stage an input spoken sentence is decomposed into one or more appropriate morphemes by Japanese morphological analyzer JUMAN [13]. An example of input/output of JUMAN is shown in Table 3.6. JUMAN adds to each morpheme several kinds of information; the original form, the name of part of speech (POS), and the name of inflection form. Note that the grammatical terminology in this article follows Masuoka-Takubo grammar [10].

⟨Feature extraction stage⟩

In this stage a feature vector for CI type determination and one for utterance proposition construction are extracted separately from the output of JUMAN [14][15]. An example of input/output in this stage is shown in Table 3.7.

A feature vector for CI type determination consists of four elements: "semantic words," names of POSs, names of inflection forms, and original forms. A semantic word is a symbol representing a group of words that similarly influence CI type determination. Semantic words were defined according to the results of analyzing 298 sentences. These words are listed in Table 3.8.

Names of POSs are extracted from nouns, demonstratives, and assertive words, and names of inflection forms are extracted from action verbs. Original forms are extracted from postpositional words introducing a subject, such as "とは ("towa")," "って ("tte")," and "は ("wa")," from the postpositional word "で ("de")" representing case, and from the postpositional words "も ("mo")" and "でも ("demo")" which are used for taking up a matter. The verbs "する (do)" and "できる (can)," the verbal postfixes "れる ("reru"), " "いる ("iru")," "る ("ru")," "せる ("seru")," and "ます ("masu")," and the adjectival postfixes "ない (not)" and "たい (want)" change a feature value extracted just before into another feature value. The rules of changes are listed in Table 3.9. When the feature value to be changed is the name of an inflection form and the new feature value is a semantic word, the feature value "action_predicate" just before the changed one is removed.

A feature vector for utterance proposition construction consists of two elements: original forms and names of parts of speech. The values of these elements are extracted mainly from nouns, verbs, adjectives, and postpositional words representing case.

⟨CI type determination stage⟩

A CI type is determined by matching a feature vector for CI type determination with the CI type determination rules listed in Table 3.10. To obtain these rules, the feature vectors for CI type determination were extracted from the 298 sentences for analysis and were arranged in the following way:

(1) The priority was attached to each semantic word taking the degree of its influence to CI type determination into consideration. This means that the higher priority a semantic word has, the smaller kinds of CI types the spoken sentences from which its semantic word is extracted have. In Table 3.8, a semantic word with a lower number has a higher priority.

(2) The following operations were done from the semantic word with a higher priority.

(2-1) The sentences from which the target semantic word was extracted were collected. These collected sentences were dealt as the initial set in the next operation (2-2). One or more kinds of CI types were observed from the set.

(2-2) For the feature vectors extracted from the sentences in the set, the following operation was repeated. A combination of the target semantic word and one or more feature values was searched for from the feature vectors. This combination was extracted from all or some of the sentences with the same CI type but was not extracted from any of the sentences with other CI types. Especially, the combination was searched for preferentially before and behind the target semantic word and near the end of the sentence. The pair consisting of the combination searched for and its CI type was registered as a rule, and the sentences from which that combination was extracted were removed out of the set. This operation was terminated when only one kind of CI type was observed from the sentences still remaining in the set.

(2-3) The pair consisting of the target semantic word and the remaining CI type was registered as a rule.

The structure of a CI type determination rule is also described in the extend Backus form.

Table 3.8 Semantic words.

	Semantic words	Examples of the corresponding words or phrases
1	dialogue_start	すみません (excuse me), あの (well)
2	dialogue_end	わかりました (I got it)
3	how_phrase	どうする, どうやる, どう (how)
4	instruction_predicate	教える (teach)
5	benefit_predicate	もらう, くれる
6	want_predicate	欲しい (want)
7	interrogative_word	誰 (who), 何処 (where)
8	what	何 (what)
9	OK_predicate	いい, よい, よろしい (alright)
10	lack_predicate	忘れる (forget), わからない (I don't know)
11	prohibit_predicate	駄目だ, いけない (must not)
12	pp_word_interrogation	Postpositional words representing interrogation like か ("ka"), ね ("ne") and よね ("yone")
13	end_of_sentence	(end of sentence)
14	method_word	仕方, 方 (how to)
15	"suru"	する (do)
16	"naNdemo"	何でも (any)
17	action_predicate	読む (read), 消す (delete), 押す (push)
18	state_predicate	State verbs like ある (be) and 違う (differ) Adjectives like おかしい (strange) and ない (be not) Verbs with the potential mood like 読める (can read) and 消せる (can delete) Verbs without intention like 消える (disappear) and 出る (appear)
19	knowledge_predicate	知る (know), 覚える (learn), わかる (find)

$$
\begin{aligned}
\langle\text{rule}\rangle &::= (\langle\text{conditional part}\rangle\ \langle\text{CI type}\rangle)\\
\langle\text{conditional part}\rangle &::= (\langle\text{pattern}\rangle\ \langle\text{pattern}\rangle^*)\\
\langle\text{pattern}\rangle &::= (\langle\text{feature value}\rangle\ \langle\text{feature value}\rangle^*)\\
\langle\text{feature value}\rangle &::= \langle\text{semantic word}\rangle \mid \langle\text{name-of POS}\rangle \mid\\
&\quad\ \langle\text{name of inflection form}\rangle \mid \langle\text{original form}\rangle
\end{aligned}
$$

How to apply the rules is shown below.

【Applying rule No. r (r = 1, 2, ⋯, R) for a feature vector for CI type determination】 From rule No. 1 at the top line in Table 3.10 to rule No. R at the bottom line, each rule is applied in order for a feature vector for CI type determination. If the conditional part of the rule was successfully matched with the feature vector, the corresponding CI type is fired and this application is terminated.

【Matching conditional part with feature vector】 This matching proceeds from the right pattern to the left pattern of the conditional part. If the target pattern matched a partial sequence of the feature vector, the matching of the pattern is successful. The patterns at the left of its partial sequence then form a new feature vector for CI type determination, and the matching of the next pattern is done with the new feature vector. If all the patterns were successfully matched, the matching of the conditional part is successful.

Table 3.9 The rules for changes of feature values by postfixes.

Changed values	Input morphemes	New values
"action noun"	"する (do)"	"action_predicate" + (inflection form of its input)
"action noun"	"できる (can)"	"state_predicate"
"knowledge_predicate"	"ない (do not)"	"lack_predicate"
(inflection form)	"たい (want)"	"want_predicate
"past form modifying a predicate"	"ます ("masu")"	"state_predicate"
(inflection form)	"ます ("masu")"	(inflection form of its input)
(inflection form)	"せる ("seru")"	(inflection form of its input)
(inflection form)	"ない (do not)"	"state_predicate"
(inflection form)	"いる ("iru")"	"state_predicate"
(inflection form)	"れる ("reru")"	"state_predicate"
(inflection form)	"る ("ru")"	"state_predicate"

⟨Utterance proposition construction stage⟩

This stage corresponds to a semantic analysis method. Although the conventional semantic analysis methods construct a semantic structure from a spoken sentence as a whole, the stage constructs a semantic structure from the feature vector for utterance proposition construction. This semantic structure is an utterance proposition and is described in frame representation. Each frame is generated from a pair consisting of an original form and a name of POS in the feature vector and is bound by one of other frames if possible.

Consider spoken sentence "終了を押せばいいんですね (Should I push the Close window button?)" as an example. In the morphological analysis and feature extraction stages, the feature vector for utterance proposition construction is extracted from the sentence, which is a sequence of "終了 (Close window)," "action noun," "を ("wo")," "postpositional word representing case," "押す (push)," and "verb." The object frame shown in Fig. 3.1(b) is generated from the feature values "終了 (Close window)" and "action noun," where the value "終了 (Close window)" is bound in the name slot in the object frame. The postpositional word representing case, "を ("wo")," is bound in the case slot in the object frame. An action frame representing pushing something is generated from the feature values "押す (push)" and "verb." The above object frame is bound in the target slot in this action frame. As the result, the action frame is specialized to that shown in Fig. 3.1(a). This specialized action frame is the utterance proposition obtained from the example sentence.

If there is no ambiguity in relation between frames, case information is not used. Otherwise, case information is used for the binding of slots of frames. Consider the act "メッセージを inbox から myfolder に移す (moving a message from folder "inbox" to folder "myfolder")." The "メッセージ (message)" object frame, the "フォルダ (folder)" object frame whose name slot value is "inbox," and the "フォルダ (folder)" object frame whose name slot value is "myfolder" are respectively bound in the target slot, the starting folder slot, and the destination folder slot in the "メッセージ移動 (MoveMessage)" action frame. This results from the knowledge that the "フォルダ (folder)" object frames whose case slot values are "から (from)" and "へ (to)/に (to)" can respectively be bound in the starting folder slot and the destination folder slot of the "メッセージ移動 (MoveMessage)" action frame. Postpositional words representing case are often omitted or used wrongly by users. However, when case information is needed, it is used correctly [16].

Table 3.10 CI type determination rules.

(((dialogue_start))	d-start)
(((dialogue_end))	d-end)
(((how_phrase))	ask-how)
(((instruction_predicate))	ask-wh)
(((benefit_predicate))	have-goal)
(((want_predicate))	have-goal)
(((interrogative)(OK_predicate))	ask-wh:OK)
(((interrogative)(basic form end_of_sentence))	ask-wh:OK)
(((interrogative))	ask-wh)
(((what も)(OK_predicate))	ask-if:OK)
(((what も))	have-belief)
(((とは what))	ask-about)
(((って what))	ask-about)
(((what assertive word))	ask-how)
(((what "suru"))	ask-how)
(((what))	ask-wh:OK)
(((state_predicate OK_predicate))	ask-if:OK)
(((action_predicate)(OK_predicate))	ask-if:OK)
(((OK_predicate))	ask-if:EQ)
(((common noun lack_predicate))	ask-wh)
(((state_predicate lack_predicate))	ask-wh)
(((action noun lack_predicate))	ask-about)
(((pp_word_interrogation lack_predicate))	ask-if)
(((lack_predicate))	ask-how)
(((prohibit_predicate))	ask-if:NO)
(((basic form pp_word_interrogation))	ask-if:OK)
(((pp_word_interrogation))	ask-if)
(((basic form end_of_sentence))	ask-if:OK)
(((end_of_sentence))	have-belief)

⟨Construction of CI description⟩

A CI description consists of the outputs of the CI type determination and utterance proposition construction stages. Consider the example sentence "終了を押せばいいんですね (Should I push the Close window button?)" again. As a feature vector for CI type determination, the sequence "action noun, action_predicate, conditional form, OK_predicate, pp_word_interrogation, end_of_sentence" is extracted. The CI type "ask-if:OK" is selected according to the CI type determination rules. As a feature vector for utterance proposition construction, the sequence "終了 (Close window), action noun, を ("wo"), postpositional word representing case, 押す (push), verb" is extracted. Then the action frame #⟨PushCommandButton #X139EG76⟩ is generated from the sequence. Hence, the CI description constructed from the example sentence is

(ask-if:OK #⟨PushCommandButton #X139EG76⟩),

where the structure of #⟨PushCommandButton #X139EG76⟩ is shown in Fig. 3.1.

Table 3.11 Accuracy of CI type determination.

	Accuracy(%)
298 sentence for analysis	100.0
177 sentences for evaluation	89.3

Table 3.12 The newly added CI type determination rules.

(((method_word instruction_predicate))	ask-how)
((("naNdemo" OK_predicate))	ask-if:EQ)
(((interrogative でも)(OK_predicate))	ask-if:EQ)
(((interrogative)(basic form pp_word_interrogation))	ask-wh:OK)
(((what)(も)(OK_predicate))	ask-if:OK)
(((what も)(pp_word_interrogation))	ask-if)
(((common noun は what assertive word))	ask-wh)
(((what state_predicate))	ask-wh)
(((demonstrative pronoun で OK_predicate))	ask-if:EQ)
(((adverbial noun で OK_predicate))	ask-if:EQ)
(((action noun で OK_predicate))	ask-if:OK)
(((common noun state_predicate lack_predicate))	ask-how)

3.5 Evaluation of the CI Type Determination Method

This subsection uses the 177 evaluation sentences to evaluate the proposed CI type determination method. The CI types of all the 475 sentences were first determined manually by us on the basis of the criteria listed in Tables 3.3 and 3.5. Then the 29 CI type determination rules listed in Table 3.10 were obtained from the 298 sentences for analysis in the way described in subsection 3.4. These rules were used in the following experiments.

The morphological analysis, feature extraction, and CI type determination were done in these experiments. In the first stage, an input spoken sentence was decomposed by JUMAN and the output of JUMAN was passed to the next stage. In the feature extraction stage, a feature vector for CI type determination was extracted from the output of JUMAN and was passed to the final stage. In the CI type determination stage, a CI type was determined by applying the rules shown in Table 3.10 for the feature vector. If this CI type is the same type with the predetermined CI type, the CI type determination is correct. Otherwise, it is wrong. The results of the experiments are listed in Table 3.11. The computer used in the experiments was a Sun SPARC Station IPX (SunOS 4.1.3, memory size 32 MB).

The CI type of a spoken sentence was determined by using only the surface knowledge, which includes original forms, names of POSs, and names of inflection forms, and simple semantic information implemented as semantic words. However, the CI types of all of the 298 sentences for analysis were determined correctly. This means that the information extracted from a spoken sentence is sufficient for determining the CI type of its sentence. The CI types of about 90% of the 177 sentences for evaluation were determined correctly although only 29 rules were used for determining a CI type. Thus the CI type determination method is expected to be highly accurate even for unlearned data since the CI type determination rules were not obtained from the 177 sentences for evaluation.

Table 3.13 Ten rules used most frequently.

Rules		# of rules
(((how_phrase))	ask-how)	114
(((pp_word_interrogation))	ask-if)	72
(((end_of_sentence))	have-belief)	42
(((action_predicate)(OK_predicate))	ask-if:OK)	38
(((dialogue_start))	d-start)	34
(((basic form pp_word_interrogation))	ask-if:OK)	28
(((demonstrative pronoun で OK_predicate))	ask-if:EQ)	20
(((interrogative))	ask-wh)	16
(((lack_predicate))	ask-how)	13
(((want predicate))	have-goal)	8

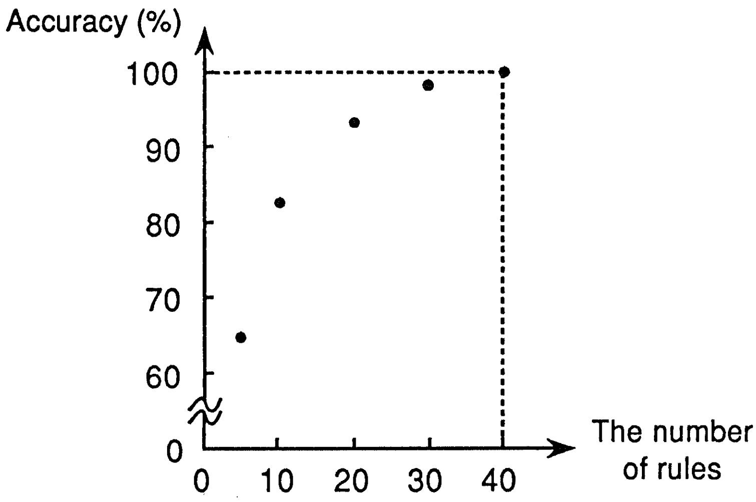


Figure 3.3 The accuracy of determining a CI type.

The data sets for evaluation were then transferred to those for analysis one by one, and the change of the accuracy of CI type determination was investigated. The CI types of all the 28 sentences of two data sets for evaluation were determined correctly by using 41 rules obtained by analyzing 447 sentences of 14 data sets. The 12 added rules are listed in Table 3.12.

The 41 rules listed in Tables 3.10 and 3.12 were ranked in order of frequency of use for all the 475 sentences, and the 10 rules used most frequently are listed in Table 3.13 with their frequencies. The relation between the number of rules and the accuracy of the CI type determination was also investigated for all the 475 sentences, where the rules were applied from the rule used frequently. These results are shown in Fig. 3.3.

Figure 3.3 shows that the 41 rules can be classified into the rules applicable to many sentences and ones applicable to a few or only one sentences. If the 20 rules of highest applicability are used in CI type determination, the method is expected to be accurate for more than 90% of unlearned input sentences.

3.6 Characteristics of the CI Description and CI Recognition Method

The communicative intention of a spoken sentence is described as a pair consisting of a CI type and an utterance proposition. The utterance contents expressible in this description form are restricted because a description of CI is given as a complex of modality information and propositional information. The utterance contents are restricted to the objects or events in the task domain. A reflective expression, for example, "one knows what one knows," cannot be expressed in this description form. Because the CI type of a sentence must be selected out of at most 12 different symbols, the support system cannot correctly interpret spoken sentences that do not correspond to any of the pre-defined 12 CI types. In an advisory environment, however, these restrictions are allowable.

Kume et al. [17] defined seven illocutionary force types for machine translation systems in the "inquiry on a conference" domain. In their type determination method, the heuristic rules formulated from knowledge on illocutionary acts in the original language are used together with such global constraints as the end expression of a sentence and the use of adverbs. The machine translation can be done at the level of communicative intention by using knowledge on illocutionary acts in the target language. Their goal is the machine translation of spoken language, but their method cannot cope with the illegal sentences appearing in actual spoken dialogues because their method is based on general pragmatics.

Iida et al. [18] defined about 20 types of communicative act from the viewpoint of the roles of utterances in a dialogue. Type is determined by using the modality information of a sentence, and the determined type is used as an element of an utterance pair. Their research was focused not on the type determination but on the use of the determined type, which was used as a key element for constructing utterance pairs in dialogue understanding. Their works did not explain their type determination method.

Den et al. [4] presented a semantic analysis method for conversational sentences. This method can resolve the ellipsis observed in conversational sentences. The method restricts the number of interpretation candidates of a sentence by using the idea of abduction [19]. The method, however, corresponds to the utterance proposition construction stage and does not deal with the modality information of a sentence.

In this section, the relations between combinations of feature values for CI type determination and 12 different CI types were formulated statistically and implemented as the CI type determination rules. In other words, the rules are regarded as the constraints between the surface structures of spoken sentences and the types of the CI represented by the sentences. Consider the example sentence "メッセージを送るにはどうしたらいいんですか (What should I do to send a message?)." This sentence is decomposed into three phrases: "メッセージを送る (send a message)," "どうしたら (What do I do)," and "いいんですか (should I)." The sequence "common noun, action_predicate, basic form, には ("niwa"), how_phrase, OK_predicate, pp_word_interrogation, end_of_sentence" is extracted from the sentence as a feature vector for CI type determination, and then the CI type "ask-how" is selected according to the CI type determination rules because the feature value "how_phrase" is included in the vector. The action frame "SendMessage" is generated from the sequence "メッセージ (message), common noun, を ("wo"), postpositional word representing case, 送る (send), verb." This sequence is extracted from the sentence as a feature vector for utterance proposition construction. As mentioned above, the method is robust and is accurate even when used for the illegal or ungrammatical expressions often occurring in spoken language. Note, however, that the target spoken sentences are restricted to sentences for asking a consultant for help.

The method has two features. One is that the feature values for CI type determination are derived only from the surface knowledge of a sentence, which includes original forms, names of POSs, and names of inflection forms, and simple semantic information implemented as semantic words. The other feature is that the CI type of a sentence is determined by matching an input feature vector for CI type determination with at most 41 rules. This enables CI type to be determined faster than an utterance proposition can be constructed.

3.7 Conclusion

In this section, a method was developed to recognize communicative intention (CI) from a user utterance based on the analysis of real user utterances. The recognized CI is represented by a CI description consisting of a CI type and an utterance proposition. The propositional information representing actions, states, and objects in the task domain is extracted from an utterance, and is described as an utterance proposition in frame representation, and the modality information representing the user's attitude type to the corresponding utterance proposition is extracted from the utterance, and is described as one of the 12 different symbols shown in subsubsection 3.3.1.

A description form of CI was defined from the viewpoint of the roles of user utterances in an advisory dialogue. For this reason, only information necessary to request help can be described in this description form. Hence, the CI recognition method is robust and can be used even for the illegal or ungrammatical expressions which are often observed in spoken language.

Utterance data was obtained from the advisory dialogue database described in Section 2, which consist of 475 sentences which 16 users said to the consultant in the XMH usage experiments. The 298 sentences in the utterance data were analyzed to define a description form of CI and to develop a method for constructing a CI description from a sentence. In this article, especially, a method for determining the CI type of a sentence was focused on.

The CI type determination method is accurate for all of the learned data consisting of the 298 sentences analyzed and for 89.3% of the unlearned data consisting of the remaining 177 sentences.

The CI recognition method deals only with a simple sentence and a complex sentence with at most one performative verb due to the insufficient specifications of the utterance proposition construction stage. The next objective is to extend the stage so as to deal with a complex sentence with two or more performative verbs.

References

[1] Kumamoto, T., Ito, A., and Ebina, T.: "Recognizing user communicative intention in a dialogue-based consultant system," *Trans. IEICE*, Vol. J77-D-II, No. 6, pp. 1114-1123 (1994) (in Japanese).

[2] Kumamoto, T., Ito, A., and Ebina, T.: "An analysis of Japanese sentences in spoken dialogue and its application to communicative intention recognition," *Proc. International Conference on Spoken Language Processing (ICSLP '94)*, Vol. 2, No. S17-19.1, pp. 943-946, Yokohama, Japan (1994).

[3] Iida, H.: "Pragmatic characteristics of natural spoken dialogues and dialogue processing issues," *Journal of JSAI*, Vol. 3, No. 4, pp. 445-452 (1988) (in Japanese).

[4] Den, Y., and Iida, H.: "An information based analysis of conversational sentences," *Technical Report of JSAI*, SIG-F/H/K-9101-12, pp. 111-120 (1991) (in Japanese).

[5] Guindon, R: "How to interface to advisory system? — users request help with a very simple language —," *Proc. CHI '88 Conference*, pp. 191-196, (1988).

[6] Allen, J. F., and Perrault, C. R.: "Analyzing intention in utterances," *Artificial Intelligence*, Vol. 15, pp. 143-178 (1980).

[7] Masuoka, T.: "Modarithi no bunpou (Grammar of modality)," Kurosio Publishers, Tokyo (1991) (in Japanese).

[8] Ito, A., and Kumamoto, T.: "A semantics for a frame representation with circular concept definition," *Trans. IPSJ*, Vol. 34, No. 8, pp. 1731-1740 (in Japanese).

[9] Ito, A., Higuchi, H., Kumamoto, T., and Ebina, T.: "A knowledge representation for a dialogue system understanding natural language and command input in the unified context," *Proc. Japan-Australia Joint Symposium of Natural Language Processing*, pp. 95-105, Iizuka, Japan (1991).

[10] Masuoka, T., and Takubo, Y.: "Kiso nihongo bunpou (basic Japanese grammar)," Kurosio Publishers, Tokyo (1991) (in Japanese).

[11] Kumamoto, T., Ito, A., and Ebina, T.: "Design and development of a dialogue-based consultant system for supporting the use of electronic mail," *Journal of the CRL*, Vol. 42, No. 2, pp. 161-171 (1995).

[12] Kumamoto, T., Ito, A., and Ebina, T.: "An experimental dialogue-based consultant system for XMH users," *Proc. the 7th Annual Conference of JSAI*, 20-3, pp. 555-558 (1993) (in Japanese).

[13] Myoki, Y., Matsumoto, Y., and Nagao, M.: "User customizable Japanese dictionary system and morphological analyzer," *Proc. the 42nd Annual Meetings of IPSJ*, 1C-9, pp. 3-17 - 3-18 (1991) (in Japanese).

[14] Kumamoto, T., Ito, A., and Ebina, T.: "Improved communicative intention recognition in spoken Japanese sentences by extracting propositional and modality information separately," *Proc. of International Conference on Computer Processing of Oriental Languages (ICCPOL '95)*, pp. 68-75, Hawaii, U.S.A. (1995).

[15] Kumamoto, T., Ito, A., and Ebina, T.: "On a communicative intention recognition method extracting propositional information and modality information separately from a spoken sentence," *Technical Report of JSAI*, SIG-SLUD-9403-2, pp. 7-12 (1995) (in Japanese).

[16] Kumamoto, T., Ito, A., and Ebina, T.: "On speech understanding in a dialogue-based consultant system for computer users," *Record of the 1992 Kansai-Section Joint Convention of Institutes of Electrical Engineering*, G8-10, p. G268 (1992) (in Japanese).

[17] Kume, M., and Kogure, K.: "Inference mechanism of illocutionary force types for MT systems," *Proc. the 39th Annual Meeting of IPSJ*, 3F-5, pp. 612-613 (1989) (in Japanese).

[18] Iida, H., and Arita, H.: "Natural language dialogue understanding on a four-typed plan recognition model," *Trans. IPSJ*, Vol. 31, No. 6, pp. 810-821 (1990) (in Japanese).

[19] Hobbs, J. R., Stickel, M., Martin, P., and Douglas Edwards: "Interpretation as abduction," *Proc. the 26th Annual Meeting of the Association for Computational Linguistics*, pp. 95-103 (1988).

4. Control and Management of Dialogues with a User

In this section, real user-consultant dialogues are analyzed, and a dialogue processing method adapted to the characteristics of user-consultant dialogues is developed based on the analysis [1][2]. This method can manage a dialogue between a user and a support system, the structure of which is a complex of dialogue patterns, and can guide a user to accomplish tasks by controlling system responses. Hence, the method has been implemented into the support system [3][4] we are developing.

4.1 Introduction

A dialogue between a user and a support system is goal-oriented, and is conducted in order for an XMH user to accomplish tasks. Hence, the dialogue has a complex structure; a dialogue may have one or more subdialogues, each of which is conducted to help the user in achieving his subgoal. Similarly, a subdialogue may have one or more sub-subdialogues, each of which is conducted to help the user in achieving his sub-subgoal. And the dialogue should be controlled so that the user can achieve his task goal effectively.

Conventional studies on dialogue processing are generally classified into ones [5]-[7] to understand a dialogue and ones [8] to control a dialogue. Studies by Iida et al., Takano et al. and Ohira et al. are classified into the former studies. Studies by Nishiyama et al. are classified into the latter studies.

Iida et al. [5] and Takano et al. [6] proposed methods for understanding a dialogue between people based on the idea of "utterance pair." Although an utterance pair corresponds basically to a pair consisting of a question and a response, it was originally extended by each research group to understand a dialogue between people. Ohira et al. [7] proposed a method for managing a dialogue between a user and a dialogue system in database retrieval. This method was also based on the idea of "utterance pair." These methods were developed to understand or manage a dialogue with a complex structure. Hence, they were not designed to guide a user to achieve his task goals by controlling system responses. Basically, the "utterance pair" mechanism is not suitable for dialogue control.

The target dialogue of Nishiyama et al. [8] also was a dialogue between a user and a dialogue system in database retrieval, but they proposed a method for not managing but controlling a dialogue. This method can guide the user to the target data effectively by controlling the system responses. However, the method is not available in other task domains because an utterance which should produce a subdialogue is interpreted as change, addition or deletion of retrieval condition and a subdialogue is not produced.

The approach of this research is based on an analysis of real data [9][10]. Hence, real dialogue data was obtained from the advisory dialogue database described in Sect.**2**. The dialogue data consists of advisory dialogues between 16 naive users and an expert consultant in the XMH usage experiments shown in Sect.**2**. Based on an analysis of the dialogue data, a dialogue processing method was developed which behaves like a human consultant. This method can manage a dialogue between a user and a support system, the structure of which is a complex of dialogue patterns, and can guide a user to accomplish tasks effectively by controlling system responses.

This section is organized as follows. Subsection **4.2** describes dialogue data obtained from the advisory dialogue database. Subsection **4.3** analyzes the dialogue data, and obtains several findings to develop the dialogue processing method. Subsection 4.4 determines requirements from the findings, and proposes a dialogue processing method which satisfies the requirements. Subsection **4.5** concludes this section and describes our plans for future work in dialogue processing.

Table 4.1 Statistical summary of the collected dialogue data.

User utterances		499 utterances (31.2 utterances per user)
User actions	Mouse operations	3,495 operations (218.4 operations per user)
	Key operations	12,796 keys (799.8 keys per user)
	Guided operations	318 operations (19.9 operations per user)
Consultant responses		652 utterances (40.8 utterances per session)

4.2 Dialogues in the Advisory Dialogue Database

In this subsection, specifications of the dialogue data to be analyzed are shown.

The dialogue data was obtained from the advisory dialogue database described in Sect.**2**. They consist of advisory dialogues between 16 naive users and an expert consultant in the XMH usage experiments shown in Sect.**2**. Table 4.1 statistically summarizes the dialogue data. In Table 4.1, the number of mouse operations means the number of times the users pushed a button on the mouse, and the number of key operations means the number of times the users hit a key on the keyboard. A "guided operation" means a mouse operation or a key operation which was executed by complying with the consultant's instruction, where a guided operation corresponds to pushing a mouse button once or typing a sequence of keys. A user action means a guided operation in the following.

Two sample dialogue transcripts are shown below; the symbols "U" and "C" denote a user utterance/action and a consultant response, respectively. The numbers following the symbols are serial numbers. To make it easier to follow the transcripts, user actions are in brackets. The parenthesized sentences are the English versions of the utterances.

Dialogue No. 1:

U1: どうやって移動するんですか
(How do I move a message?)

C2: はじめに移動するメッセージを選択して下さい
(First, select the target message.)

U3: [Message "4" was selected.]

C4: つぎに、移動先のフォルダを選択して下さい
(Next, select the destination folder.)

U5: [Folder "myfolder" was selected.]

C6: それから、メッセージメニューの中のメッセージの移動、を実行して下さい
(Next, execute the "MoveMessage" command on the "Message" menu.)

U7: [The "MoveMessage" command was executed.]

C8: 最後に、フォルダ目次の中の変更の実行、を実行して下さい
(Finally, execute the "CommitChanges" command on the "TableOfContents" menu.)

U9: [The "CommitChanges" command was executed.]

C10: 移動されました
(The message was moved.)

Dialogue No. 2:

U11: 手紙の内容が出てこないんですけど
(The contents of the message are not displayed.)

C12: メッセージ2を選択して下さい。
(Select message "2.")

U13: [Message "2" was selected.]

C14: つぎに、メッセージメニューの中の次のメッセージを実行して下さい
(Next, execute the "NextMessage" command on the "Message" menu.)

U15: [The "ComposeMessage" command was executed wrongly, which is next to the "NextMessage" command.]

U16: 二つ画面が出てきたんですけど
(Another window appeared.)

C17: 新しい画面の左下にある終了、を押して下さい。
(Push the "Close window" button at the bottom left-hand corner of the new window.)

U18: [The "Close window" button was pushed.]

U19: [The "NextMessage" command was executed.]

4.3 Analysis of the Dialogues

In this subsection, the dialogue data obtained in subsection **4.2** is analyzed, and several findings are obtained to determine requirements for the dialogue processing method to be developed.

The target dialogues were goal-oriented and were advisory dialogues through which a consultant helped users to use a software program. Hence, not only user utterances and consultant responses but also user actions, i.e., guided operations must be dealt with as dialogue elements. Consequently, the structure of a user-consultant dialogue is described as a sequence consisting of user utterances, consultant responses and user actions. A sequence is started by a user utterance/consultant response and is concluded by the action required or intended by the user utterance/consultant response. If another user utterance or active consultant response occurred before the current sequence is concluded, a nested sequence started by the utterance is inserted in the sequence. Here, an active consultant response is defined as a voluntary utterance which is not a response to the user.

Dialogues No. 1 and No. 2 shown in subsection **4.2** were described in the sequence form mentioned above. Then, dialogue No. 1 is the sequence "(user-utterance, consultant-utterance, user-action, consultant-utterance, user-action, consultant-utterance, user-action, consultant-utterance, user-action, consultant-utterance)." This sequence means that dialogue No. 1 has no nested sequence. Dialogue No. 2 is the sequence "(user-utterance, consultant-utterance, user-action, consultant-utterance, user-action (user-utterance, consultant-utterance, user-action) user-action)." This sequence means that dialogue No. 2 has a nested sequence; the depth of the nested sequence is one, where depth zero means the depth of a main sequence.

All the dialogue data obtained in subsection **4.2** were described in the sequence form. As results, 317 main sequences were obtained; 222 of them had no nested sequence and the remaining 95, which account for about 30.0% of all the main sequences, had one or more nested

Table 4.2 Structures of the dialogues.

(A)	222
(A_1 (B) A_2)	34
(A (B))	29
(A (B_1 (C) B_2))	6
(A (B (C)))	4
(A_1 (B) A_2 (C) A_3)	2
(A (B) (C))	2
(A_1 (B) (C) (D) A_2)	2
Other structures ((A_1 (B) A_2 (C)), (A_1 (B) A_2 (C (D)) (E) A_3), (A_1 (B) A_2 (C_1 (D) C_2)), and so on.)	1 in each (16 in total)
Total	317

sequences. However, these nested sequences did not cross another sequence. Consequently, they can be managed by using a stack.

All 317 main sequences were classified on structural basis. The results are shown in Table 4.2. An alphabetic symbol in a sequence means one or more successive dialogue elements at the same depth. Subscripts are serial numbers. For example, dialogue No. 1 shown in subsection 4.2 is described in the form of "(A)" and dialogue No. 2 is "(A_1 (B) A_2)." Note that depth six was the maximum and depth 1.4 was the average; this average was calculated by (Σ the maximum depth in each main sequence)/(the number of main sequences with one or more nested sequences).

To analyze the patterns of dialogue elements at the same depth in a sequence, all the sequences were decomposed on the basis of depth. For example, the sequence (A_1 (B) A_2) is decomposed into two sequences: "(A_1 A_2)" and "(B)." A decomposed sequence is called a "layer" in the following. From this decomposition, 265 layers were obtained from the 95 main sequences with one or more nested sequences. Adding the 222 main sequences with no nested sequence, a total of 487 layers were obtained.

A layer is expected to have dialogue patterns. To examine this, consultant responses were classified into "active utterances" and "passive utterances;" "passive utterances" were further classified into "Instruction," "Yes/No," "Attribute-value," "Explanation," "Query," and "Completion;" and user actions were classified into "Execution" and "Misoperation." All the layers were described using these detailed dialogue elements. The results are shown in Table 4.3. The denotation "[Instruction Execution]" means that a consultant instruction and a user execution were repeatedly found. An "active support dialogue" means a layer beginning with an active consultant response. Since the support system we are developing has no active support capabilities, the active support dialogues were not analyzed.

Table 4.3 shows that several typical dialogue patterns were observed as the structures of layers. The dialogue patterns of a layer can be determined by the type of communicative intention (CI) of the utterance at the beginning of the layer. The following interactions between a user and a consultant seem very natural: if the user asked a question concerning a procedure, the consultant will instruct the user in the first step of the procedure. The user will therefore comply with the instruction and execute it.

Next, the relation between the dialogue patterns and the CI types was determined. The

Table 4.3 Structures of layers (dialogues at the same depth).

(User-utterance [Instruction Execution])	134 (32.4%)
(User-utterance Yes/No)	110 (26.6%)
(User-utterance Attribute-value)	45 (10.9%)
(User-utterance Yes/No Attribute-value)	9 (2.2%)
(User-utterance Explanation [Instruction Execution])	9 (2.2%)
(User-utterance Yes/No [Instruction Execution])	8 (1.9%)
Other	98 (23.7%)
Subtotal	413 (100.0%)
Active support dialogue	28
(User-utterance)	46
Subtotal	74
Total	487

Table 4.4 Classification of user's communicative intention.

CI Types	Meanings of utterances	no. of layers
ask-how	wh-questions on procedures	146 (35.4%)
ask-about	wh-questions on concepts	7 (1.7%)
ask-wh	wh-questions on attribute values	37 (9.0%)
ask-if	yes/no-questions on truth values	179 (43.3%)
have-belief	expressions of beliefs	36 (8.7%)
have-goal	expressions of task goals	8 (1.9%)
Total		413 (100%)

CI types observed in the user utterances are shown in Table 4.4 with the meanings of the corresponding utterances. The numbers in Table 4.4 are the results of classifying the 413 layers in Table 4.3 according to the CI types.

To analyze what dialogue patterns existed in layers beginning with the same CI type, user-utterances in all the layers were replaced with CI types of the utterances. In this paper, consider the layers beginning with CI type "ask-if" or "ask-how," which account for 78.7% of all the 413 layers. The dialogue patterns for "ask-if" are shown in Table 4.5; 149 layers, which account for 83.2% of all the 179 layers, have structures of "(ask-if Yes/No)," "(ask-if Yes/No {[Instruction Execution] | Attribute-value})," or "(ask-if {[Instruction Execution] | Attribute-value})". The denotation "{A | B}" means that either A or B occurred. The dialogue patterns for "ask-how" are shown in Table 4.6; 88 layers, which account for 60.3% of all the 146 layers, have a simple structure of "(ask-how [Instruction Execution])".

The above findings show that most of the layers have typical dialogue patterns which can be determined by the CI type of the user utterance. Conversely, however, 30 layers (16.8%) of the layers beginning with "ask-if" and 58 layers (39.7%) of the layers beginning with "ask-how"

Table 4.5 Layers beginning with CI type "ask-if."

(ask-if Yes/No)	107 (59.8%)
(ask-if [Instruction Execution])	14 (7.8%)
(ask-if Attribute-value)	11 (6.1%)
(ask-if Yes/No Attribute-value)	9 (5.0%)
(ask-if Yes/No [Instruction Execution])	8 (4.5%)
Other	30 (16.8%)
Total	179 (100%)

Table 4.6 Layers beginning with CI type "ask-how."

(ask-how [Instruction Execution])	88 (60.3%)
(ask-how Attribute-value)	10 (6.8%)
(ask-how Instruction)	6 (4.1%)
(ask-how Explanation [Instruction Execution])	4 (2.7%)
(ask-how [Instruction Execution] Execution)	4 (2.7%)
(ask-how Instruction Explanation Execution)	3 (2.1%)
(ask-how Completion)	3 (2.1%)
Other	28 (19.2%)
Total	146 (100%)

did not fit the above dialogue patterns. To find why they did not fit the dialogue patterns, the structures of these layers were investigated.

First, consider the 30 layers beginning with "ask-if." 13 layers (43.3%) of the layers had atypical dialogue patterns because the consultant additionally gave supplemental explanation. 10 layers (33.3%) of them had atypical dialogue patterns in which the last dialogue element was Instruction, Explanation, Query, or Misoperation. The reasons these atypical dialogue patterns occurred were the same as the reasons the layers with illegal patterns "(user-utterance)" occurred. The reasons will be revealed later. The remaining 7 layers (23.3%) resulted from the fact that the users executed correct operations before the consultant's instruction.

Next, consider the 58 layers beginning with "ask-how." 32 layers (55.2%) of the layers occurred because the consultant additionally gave supplemental explanation. The last dialogue element in 18 layers (31.0%) of them was Instruction, Explanation, Query, or Misoperation. The reasons the last dialogue element was any of these were the same as the reasons mentioned above. 5 layers (8.6%) of them occurred because the users executed correct operations before the consultant's instruction. The remaining 3 layers (5.2%) have the structure "(ask-how Completion)."

The consultant often gave supplemental explanation according to the user's comprehension and mental state. A mechanism for generating a response like this cannot be achieved by current technology because beliefs, abilities and mental states of users can not be managed perfectly and quantitatively by computer-based systems. Such a mechanism will be developed in the future. Another complicating factor is that users often executed correct operations in-

Table 4.7 The reasons "(user-utterance)" occurs.

Additional utterance	21 (45.7%)
Consultant did not reply	14 (30.4%)
Different wording	8 (17.4%)
Other	3 (6.5%)
Total	46 (100.0%)

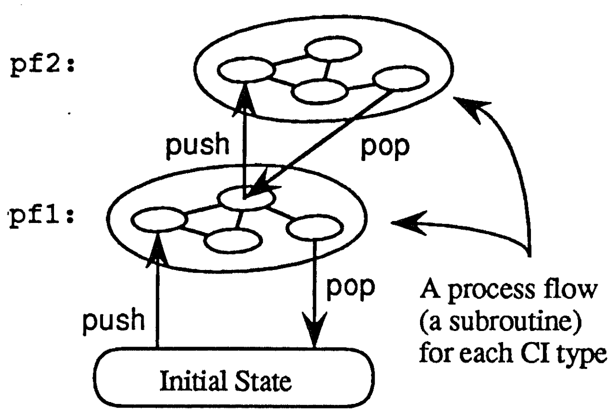


Figure 4.1 Basic concept of a dialogue processing method.

dependently of the consultant's instruction, even though the user had asked the consultant a question concerning the operations. The ability to re-evaluate the progress of the user's task plan whenever necessary must be therefore given to the support system.

Finally, the layers with illegal patterns "(user-utterance)" were analyzed. Table 4.7 shows the reasons for such patterns. An "additional utterance" is carried out to add information to the preceding utterance. Hence some relationship exists between the two utterances, for example, further specification or query. To handle these additional utterances, a new semantic representation that can describe the relationship between the two utterances must be defined. This becomes one of our future objectives.

A "different wording" means that the current utterance changed the wording of a previous utterance in the same dialogue. If the current utterance can be unified with a previous utterance in the same dialogue, the current utterance is regarded as a "different wording." Since the two utterances have complementary information to each other, the total information is described by unifying their semantic structures. The layers started by utterances between the two unified utterances are concluded as the side effect on the unification, and the layer started by the previous utterance is concluded when the layer started by the current utterance is concluded.

In this subsection, user-consultant dialogues were analyzed, and several findings were obtained to determine requirements for the dialogue processing method.

4.4 Dialogue Processing Method

In this subsection, the requirements for the dialogue processing method are determined from the findings obtained in subsection **4.3**, and the method is developed so as to satisfy these requirements. In subsubsection **4.4.1**, a basic concept of the method is designed. In subsubsection **4.4.2**, process flows are designed. A process flow controls a layer, and is described as an algorithm.

4.4.1 Design of basic concept of the method

First, the findings obtained in subsection **4.3** are described, and the requirements for a basic concept of the dialogue processing method are determined based on the findings.

(1) The target dialogue has not only user utterances and consultant responses but also user actions in XMH as dialogue elements. Consequently, the method to be developed must deal with three kinds of dialogue elements; user utterances, consultant responses and user actions.

(2) The dialogue is described as a sequence with zero or more nested sequences. However, nested sequences do not cross another sequence. Consequently, the method can use a stack to manage nested sequences.

(3) A layer has dialogue patterns, where a layer consists of dialogue elements at the same depth in a sequence. These patterns depend on the type of communicative intention (CI) of the utterance starting the layer. Consequently, a process flow which manages and controls a layer should be generated according to the CI type obtained from the utterance starting the layer.

Next, a basic concept of the method is designed so as to satisfy these requirements.

The designed basic concept is shown in Fig. 4.1. According to Fig. 4.1, the method before starting a dialogue is in the initial state. If a user utterance was inputted, an adequate process flow (pf1) is generated according to the CI type of the utterance and is piled up on a stack. The process flow at the top of the stack always runs, and manages and controls the current layer. If another utterance was inputted while a layer was processed, the process flow pf1 is interrupted and a new process flow (pf2) is generated according to the CI type of the new utterance and is piled up on the stack. If the new process flow pf2 finished processing, it is removed from the stack, and the process flow pf1 starts processing again. Note that the method checks whether a user utterance has different wording to any previous utterances. If true, the processing described in subsection **4.3** is carried out.

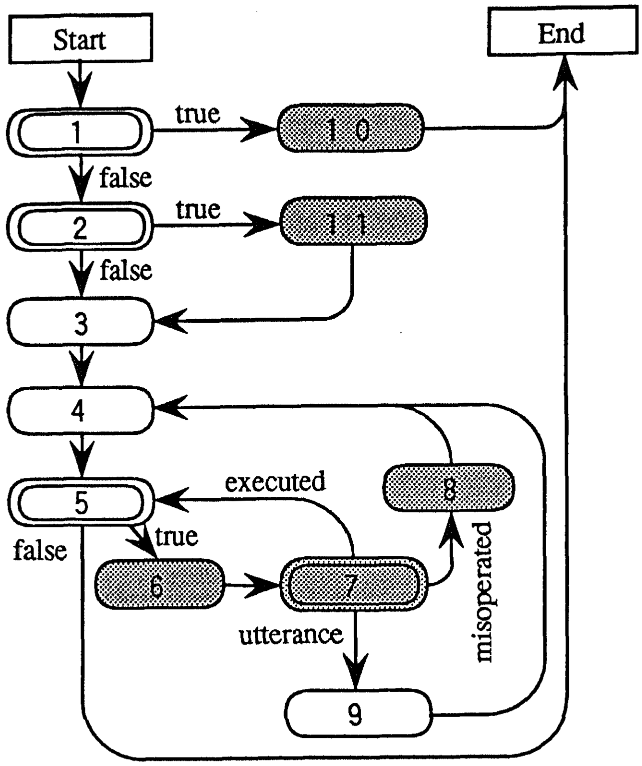
4.4.2 Design of process flows

First, the findings obtained in subsection **4.3** are described, and the requirements for process flows are determined based on these findings. Consider the design of a process flow for CI type "ask-how" in the following.

(a) Many of the layers beginning with "ask-how" have the structure "(ask-how [Instruction Execution])" as shown in Table 4.6. Consequently, a process flow for "ask-how" must have an algorithm to give adequate instructions and confirm user's executions.

(b) Users sometimes changed the state of XMH by misoperations. Consequently, a process flow for "ask-how" must have capabilities to detect a user's misoperation and inform the user of the fact, and must trace change of the state of XMH.

(c) Users sometimes asked about an operation which had already been executed successfully. In such a situation, a process flow for "ask-how" is required informing the user of the misunderstanding.



1··· tests if succeeded in operation corresponding to the utterance proposition
2··· tests if failed in operation corresponding to the utterance proposition
3··· generates a plan P1 for achieving the utterance proposition
4··· makes a new current plan from P1
5··· tests if a current plan exists
6··· instructs the first step of the current plan
7··· gets user input
8··· informs misoperation
9··· runs a new process flow
10··· informs misconception
11··· informs why failed in operation corresponding to the utterance proposition

Figure 4.2 A process flow for CI type "ask-how."

Next, a process flow for "ask-how" is designed so as to satisfy the above requirements.

The process flow for "ask-how" is illustrated in Fig. 4.2, where it is simply illustrated to make easier to follow the algorithm. Requirement (a) is satisfied by nodes No. 6 and No. 7. Requirement (b) is satisfied by nodes No. 4, No. 7 and No. 8. By node No. 7, a user input is accepted and the following conditions are checked: whether the instruction presented by node No. 6 was executed or not, or whether another utterance was inputted. If the instruction was executed correctly, a subplan by which the instruction was generated is removed from the current plan. If a wrong operation was executed, the user is informed by node No. 8, and a current plan is re-generated by node No. 4. By this plan generation, even if the state of XMH was changed by misoperations, the process flow can give adequate instructions. If a new utterance was inputted, by node No. 9 a process flow is generated according to the CI type of the new utterance and is piled up on the stack. This new process flow manages and controls a layer which was started by the new utterance, and the old process flow is stopped. If the new process flow finishes processing, the old process flow starts processing from node No. 4. By this node, a current plan is re-generated dependent on the current state of XMH. Requirement (c) is satisfied by node No. 1. By this node, whether the target operation has already been executed is checked. If done, node No. 10 informs the user.

A part of the nodes was designed based on the specifications determined in Refs.[11] and [12]. In the references, the know-how of response generation which the human consultant employed to help the users was formalized as the response generation rules. For example, the response generation rules for "ask-how" are as follows:

(i) If target operation was just before executed successfully, the user is informed.

(ii) If target operation was just before executed wrongly, the problems to be solved and their solution are presented to the user.

(iii) A plan is generated to perform the target operation, and procedures in the plan are presented to the user step by step.

These rules (i), (ii) and (iii) are achieved by nodes No. 1 and No. 10, nodes No. 2 and No. 11, and nodes Nos. 4-7, respectively.

Nodes constructing process flows are classified into four groups: 16 nodes for checking a condition, a node for getting a user input, 38 nodes for system processing, and 12 nodes for system output. In Fig. 4.2, these nodes are shown in double frames, ash-colored double frames, single frames, and ash-colored single frames, respectively.

4.5 Conclusion

A user support system is being developed which plays the role of a human consultant. Such a system should naturally and smoothly dialogue with a user and help the user to use XMH effectively. To achieve these, a dialogue processing method was developed based on the analysis of real advisory dialogues between 16 XMH users and a consultant. This method can control and manage a dialogue with a user by using process flows and a stack.

The dialogue which can be managed based on the idea of utterance pairs consists basically of pairs of a user question and a system answer, or pairs of a system question and a user answer. The dialogue processing method developed in this section can also deal with user actions, and hence it can manage and control the dialogue consisting of user questions, system responses, and user actions. Management and control are achieved by process flows and a stack. Since a process flow is described in the form of an algorithm, the response generation rules can be implemented easily into the process flow.

The characteristics of the dialogue processing method are as follows.

- ○ Not only user utterances and system responses but also user actions are included as dialogue elements. Hence, even if the user executes wrong operations, the user's misconception can be resolved by informing the user of the fact.
- ○ A dialogue consisting of one or more layers can be managed and controlled by the method. Hence, a dialogue between a user and a support system that has similar structure can also be managed and controlled naturally and smoothly.
- ○ System responses are generated so as to guide the user to achieve his task goals effectively. By this, the support system can engage effectively in a dialogue with a user.

An additional utterance is carried out to add information to the preceding utterance. To handle such an additional utterance, a method will be developed to analyze and describe a relationship between the two utterances.

References

[1] Kumamoto, T., and Ito, A.: "An analysis of user-consultant dialogue and its application to dialogue control in a dialogue-based consultant system," *Trans. IEICE*, Vol. J77-D-II, No. 8, pp. 1492-1501 (1994) (in Japanese).

[2] Kumamoto, T., and Ito, A.: "An analysis of user-consultant dialogues and its application to dialogue processing in a dialogue-based consultant system," *Systems and Computers in Japan*, Vol. 26, No. 3, pp. 103-114 (1995).

[3] Kumamoto, T., Ito, A., and Ebina, T.: "Design and development of a dialogue-based consultant system for supporting the use of electronic mail," *Journal of the CRL*, Vol. 42, No. 2, pp. 161-171 (1995).

[4] Kumamoto, T., Ito, A., and Ebina, T.: "An experimental dialogue-based consultant system for XMH users," *Proc. the 7th Annual Conference of JSAI*, 20-3, pp. 555-558 (1993) (in Japanese).

[5] Iida, H., and Arita, H.: "Natural language dialogue understanding on a four-typed plan recognition model," *Trans. IPSJ*, Vol. 31, 6, pp. 810-821 (1990) (in Japanese).

[6] Takano, A., Kashioka, H., Hirai, M., and Kitahashi, T.: "A formalization of context of dialogue and a framework for discourse analysis," *Trans. IPSJ*, Vol. 34, No. 1, pp. 88-98 (1993) (in Japanese).

[7] Ohira, E., Abe, M., Komatsu A., and Ichikawa, A.: "Flexible dialog control on information retrieval," *Trans. IEICE*, Vol. J76-D-II, No. 3, pp. 586-595 (1993) (in Japanese).

[8] Nishiyama, T., and Ooyama, Y.: "Cooperative natural language dialog processing on data base retrieval and its evaluation," *Trans. IEICE*, Vol. J73-D-II, No. 4, pp. 625-632 (1990) (in Japanese).

[9] Kawano, H., Nishio, S., and Han, J.: "Technology of knowledge discovery in databases," *Journal of JSAI*, Vol. 10, No. 1, pp. 38-44 (1995) (in Japanese).

[10] Kumamoto, T., Toraichi, K., Horiuchi, T., Yamamoto, K., and Yamada, H.: "On speeding candidate selection in handprinted Chinese character recognition," *Pattern Recognition*, Vol. 24, No. 8, pp. 793-799 (1991).

[11] Ito, A., Ebina, T., and Kumamoto, T.: "A taxonomy of the user's questions and response strategy for dialogue based user support," *Trans. IEICE*, Vol. J77-D-II, No. 7, pp. 1319-1328 (1994) (in Japanese).

[12] Ito, A., Ebina, T., and Kumamoto, T.: "Dialogue-based user support for a visual interface software — A case study for XMH —," *Proc. Pacific Rim International Conference on Artificial Intelligence (PRICAI '92)*, 1, pp. 190-196, Seoul, Korea (1992).

5. Construction of an Experimental User Support System

An experimental user support system [1][2] is constructed based on the methods developed in Sections **3** and **4**. In this section, the architecture of the support system is shown, and each module of the support system is described. Dialogue examples are also given to show how the support system works.

5.1 Introduction

Although such intelligent systems as personal computers and word processors have become widely used by the general public in Japan, they have not been used effectively because of the difficulty of their operation and the complexity of their conceptual structure. In addition, mastering the systems requires a great deal of labor because their manuals are large and complex.

In the ideal system environment, the support of an expert consultant is always available. However, such an environment does not exist for most people. Hence, a dialogue-based user support system is being developed that plays the role of an expert consultant. This support system works as follows: when a user gets into trouble while performing his task on an intelligent system, he can verbally ask the support system for help. The support system provides support to the user through spoken dialogues. The target language is spoken Japanese.

A user support system like this is experimentally constructed on a UNIX computer [1][2]. In this section, the architecture of the support system is shown, and each module of the support system is described. This article focuses on the Natural Language Processing (NLP) and Dialogue Processing (DP) modules, which correspond to the methods developed in Sections **3** and **4**, respectively.

Advisory dialogues in which the support system can participate are all related to the use of XMH as described in the previous sections. XMH is an intelligent system which has the functions of incorporating, displaying, composing, sending, and storing electronic mail, and has a visual interface operated mainly by a mouse.

The main features of the support system are summarized as follows.

(1) The XMH which is the target of support has a visual interface rather than a command interface.

UNIX Consultant (UC) by Wilensky [3], Neo-Assist by Uehara [4], and CIS (command interface shell) by Aoe [5] are conventional user support systems that help users in using intelligent systems. These support systems support intelligent systems with UNIX command interfaces rather than with visual interfaces. There are few studies addressing an intelligent system with a visual interface. User operations in a visual interface are executed mainly by mouse operations; moving a mouse cursor and pushing and releasing a mouse button. Consequently, the situation where the user has been placed in must be reconstructed from such low-level mouse operations.

(2) A CI description is constructed from a spoken sentence, where a "CI description" means a semantic structure to represent a user's communicative intention, and a "spoken sentence" means one transcribed from an utterance [6][7].

An utterance has propositional and modality information [8]. By propositional information the objective actions, states, or objects are represented which can be defined in a task domain, and by modality information the speaker's subjective judgement or attitude is represented. For example, the utterance "ファイルを消したい (I want to delete a file)" has the propositional information to represent an act of "ファイルを消す (deleting a file)" and the modality information to represent the speaker's attitude of "〜したい (wanting to do)". However, in conventional studies on spoken-language understanding [9][10], the syntactic/semantic analysis of a spoken sentence is focused on, and propositional and modality information are not distinguished from each other. Therefore, it is not easy to determine the speaker's attitude represented by modality information. On the contrary, the spoken-language understanding method, called "CI recognition method", developed in Section **3** extracts the propositional and modality information separately from a spoken sentence, and constructs a CI description which has the following structure:

$$(\langle \text{CI type} \rangle \ \langle \text{utterance proposition} \rangle),$$

where ⟨utterance proposition⟩ is a frame [11][12] which represents the propositional information of the spoken sentence, and describes the actions, states, or objects which can be defined in the XMH domain. ⟨CI type⟩ is a symbol which represents the type of the speaker's attitude to the corresponding ⟨utterance proposition⟩, and is selected out of the 12 different CI types defined in Section **3**.

(3) A system response is generated according to the response generation rules which are selected by the CI type of a user question [13][14].

User questions are classified into 12 groups from the viewpoint of what the users required of the consultant. A CI type is assigned to each group. Since the know-how when the consultant responded to the user questions with a CI type is formalized as the response generation rules for questions with the CI type, the support system can provide practical and effective support to users by using the rules prepared for each CI type.

(4) The support system controls dialogues with a user, and guides the user to his task goal [15][16].

A dialogue between a user and a support system has a dialogue pattern, such as the ordered sequence of a user question and a system response. The type of a dialogue pattern is determined by the CI type of the utterance at the beginning of the dialogue. This is because the following interactions between a user and a support system seem very natural: if the user asked a question concerning the procedures of an operation, the support system will instruct the first step of the procedures to the user. The user will therefore comply with the instruction and execute it. Hence, there are 12 different dialogue patterns which correspond to CI types one by one. A dialogue with each dialogue pattern is controlled by a "process flow" determined by the CI type of the utterance at the beginning of the dialogue. The process flow also achieves the response generation rules for the CI type, which were described in the above item (3). In the meantime, a dialogue between a user and a support system may have nested dialogues, such as subdialogues and sub-subdialogues. Since these nested dialogues do not cross another dialogue, each nested dialogue is also controlled by a process flow. Process flows for a dialogue and nested dialogues are managed by using a stack.

This section is organized as follows. Subsection **5.2** shows how a user works with the experimental user support system. Subsection **5.3** shows the architecture of the support system, and describes each module of the support system. Subsection **5.4** shows dialogues between the support system and a simulated user. Subsection **5.5** concludes this section, and shows our plans for future work on the design and construction of a dialogue-based user support system.

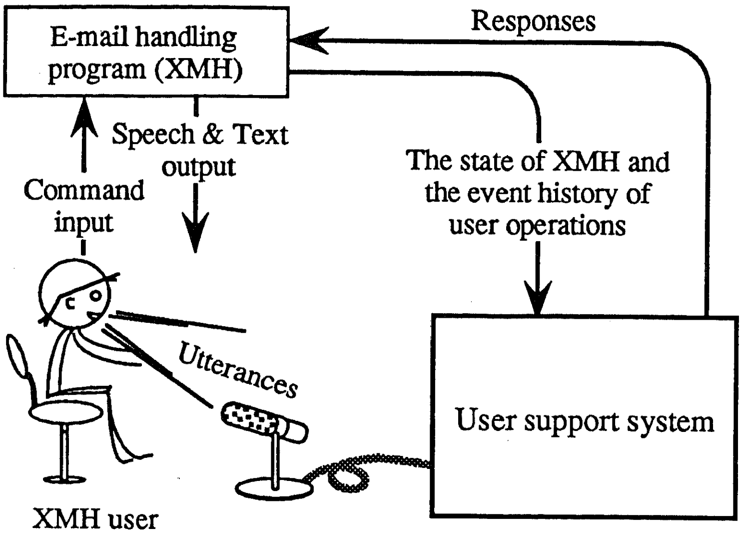


Figure 5.1 User support system environment.

5.2 User Support System Environment

The environment in which the support system is available is shown in Fig. 5.1. An XMH user usually inputs XMH commands by mouse operations, and performs his task. This is because command input by mouse operations has higher efficiency than that by natural language expressions. When a problem occurs, the user can ask the support system for help through natural language dialogues. This is because a natural language such as Japanese is very effective in expressing ambiguous concepts and/or vague requirements. It is nearly impossible in an artificial language because an artificial language has exact semantics. The support system understands the user's speech, and generates the cooperative response for the speech. The response is presented to the user in the forms of speech and text. The text form of response was kept on the user's display so that the user could refer to it at any time.

5.3 Construction of an Experimental User Support System

In this subsection, an experimental user support system [1][2] is constructed based on the methods developed in the preceding sections.

The architecture of the support system is shown in Fig. 5.2, and each module of the support system is described in the following.

Speech Recognition module:

The speech recognition system DS200 [17] on the market was incorporated into the support system as the Speech Recognition module. This DS200 transforms continuous speech by an unspecified person appropriately into one of predefined Kanji-Kana mixed sentences. The speech recognition result of an utterance is called a "spoken sentence" in this article.

Natural Language Processing (NLP) module:

This module corresponds to the CI recognition method developed in Section **3**. The module generates a CI description from a spoken sentence (a Kanji-Kana mixed sentence). A CI description is a semantic structure for describing the communicative intention which a spoken sentence has, and consists of a CI type and an utterance proposition. An utterance proposition

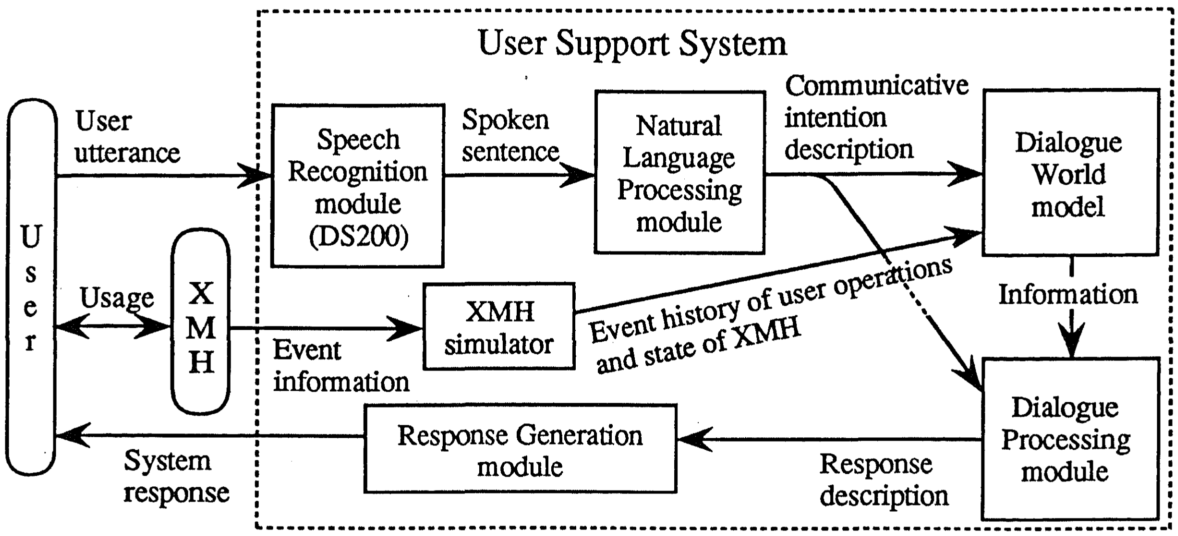


Figure 5.2 The architecture of an experimental user support system.

represents the actions, states, or objects which can be defined in the XMH domain, and is described in frame representation [11][12]. A CI type represents the user's mental attitude to the corresponding utterance proposition, and is selected out of the 12 symbols defined in Section **3**.

Dialogue World model:

This model manages the knowledge available to the support system in response generation, and consists of the Object World model [12] and the User model. The Object World model manages the event history of user operations to XMH, the state of XMH, and the static domain knowledge on XMH. The User model manages the user's beliefs and goals. An utterance proposition generated in the NLP module is described based on this Dialogue World model.

XMH simulator:

This simulator extracts event information from XMH, and simulates the user operations to XMH and the state of XMH in real time. The results of this simulation change the state of the Dialogue World model.

Dialogue Processing (DP) module:

This module corresponds to the dialogue processing method developed in Section 4, and manages and controls dialogues with a user by using process flows and a stack. A process flow is a mechanism to control a dialogue, and is executed according to the CI type of a CI description passed from the NLP module. Nested dialogues such as subdialogues and sub-subdialogues are also controlled by process flows in each. All process flows for a dialogue and nested dialogues are managed by using a stack. The process flow at the top of the stack is always running. If the process flow finishes controlling a dialogue, the process flow is removed from the stack.

If necessary, the module constructs a response description, and passes it to the Response Generation module. For example, in order to generate the response sentence "「メッセージ」メニューの「メッセージの移動」を実行して下さい. (Execute the "MoveMessage" command on the "Message" menu.)," the response description "(Instruct (menu メッセージ メッセージの移動))" is constructed and then passed to the Response Generation module.

This module also has the characteristics as follows. The response generation rules for each CI type were achieved by a process flow for the CI type in the form of an algorithm [13][14]. Hence, this module enables the support system to talk cooperatively with the user. In addition, ellipses of fillers observed in a user utterance can be resolved by using the user's task plan.

Response Generation module:

This module generates a sentence from a response description by using a template. For example, the response description (Instruct (menu メッセージ メッセージの移動)) passed from the DP module is transformed into the sentence "「メッセージ」メニューの「メッセージの移動」を実行して下さい. (Execute the "MoveMessage" command on the "Message" menu.)" by using the template "「 *1 」メニューの「 *2 」を実行して下さい. (Execute the " *2 " command on the " *1 " menu.)."

5.4 Dialogues between the Support System and a Simulated User

In this subsection, three dialogues between the experimental user support system and a simulated user are given as examples to show how the support system works. The first example is a dialogue which has a typical dialogue pattern but has no nested dialogue. The symbols "U" and "S" mean a user utterance/action and a support system response, respectively. The numbers following the symbols are serial numbers.

(Dialogue example No. 1: Dialogue with a typical dialogue pattern)

U20: メッセージ3を移すにはどうしたらいいんですか
(What should I do to move message "3"?)

S21: はじめに、メッセージ3を選択して下さい。
(First, select message "3.")

U22: [Message "3" was selected.]

S23: 次に、移動先となるフォルダを選択して下さい。
(Next, select the destination folder.)

U24: [Folder "myfolder" was selected.]

S25: 次に、「メッセージ」メニューの「メッセージの移動」を実行して下さい。
(Next, execute the "MoveMessage" command on the "Message" menu.)

U26: [The "MoveMessage" command was executed.]

S27: 最後に、「フォルダ目次」メニューの「変更の実行」を実行して下さい。
(Finally, execute the "CommitChanges" command on the "TableOfContents" menu.)

U28: [The "CommitChanges" command was executed.]

The user utterance U20 is transformed into a Kanji-Kana mixed sentence by the Speech Recognition module, and then the sentence is transformed by the NLP module into the CI description:

$$(\text{ask-how} \ \ \#\langle\text{MoveMessage} \ \#\text{X123EH56}\rangle),$$

where the #⟨MoveMessage #X123EH56⟩ is the utterance proposition which has the structure shown in Fig. 5.3 (a), the "ask-how" is the CI type which represents that the user is asking the procedures of an operation. This CI description is passed to the DP module. The DP module checks the CI type in the CI description passed from the NLP module, and executes a process flow for the "ask-how" shown in Fig. 4.2. By nodes No. 3 and No. 4 in the process flow, a current plan is generated to move message "3" into another folder. By node No. 6, a response description is generated to instruct the user to select message "3," and is then passed to the Response Generation module. The Response Generation module generates the response

#<MoveMessage #X123EH56>

target slot value is #<Message #X114EA90>

(a) An action frame representing the act of moving message "3".

#<Message #X114EA90>

name slot value is "3"

(b) An object frame representing message "3".

Figure 5.3 Frames representing an action and an object.

sentence S21 from the response description. Since the user correctly performed the instruction as shown in U22, the subplan by which the instruction was generated is removed from the current plan. Then nodes No. 5, No. 6 and No. 7 are executed repeatedly until message "3" is moved into another folder.

The second example is a nested dialogue. Suppose that the nested dialogue is produced just after S21 in dialogue example No. 1. Note that, in a spoken sentence, the words in parenthesis mean the filler which was omitted, and were not actually uttered.

(Dialogue example No. 2: Nested dialogue)

U21-1: えっとどうやって(メッセージ3を)選択するんですか
(Well, how do I select (message "3")?)

S21-2: メッセージ3の行の上にマウスカーソルを置き、マウスボタンを押して下さい。
(Put the mouse cursor on the line corresponding to message "3," and then push the mouse button.)

U21-3: [Message "3" was selected.]

S23: 次に、移動先となるフォルダを選択して下さい。
(Next, select the destination folder.)

For S21, the user said U21-1. The support system stops controlling the current dialogue when the user utterance U21-1 is inputted, and executes a process flow for the nested dialogue initiated by U21-1. This nested dialogue is concluded if the user does the action U21-3. Since the action U21-3 is equal to U22, the system response S23 is presented as the next instruction. Here, you find that the phrase "メッセージ3を (message "3")" was supplied as the direct object of the verb "選択する (select)" in U21-1 because the task goal was to move message "3".

The last example is a dialogue which has a short sentence fragment.

(Dialogue example No. 3: Dialogue containing a short sentence fragment)

U30: メッセージ3を選択したらいいんですか
(Should I select message "3"?)

S31: 何をしたいんですか？
(What do you want to do?)

U32: メッセージの移動

(To move a message.)

S21: はじめに、メッセージ3を選択して下さい。

(First, select message "3.")

After the user utterance U30 was transformed into a Kanji-Kana mixed sentence by the Speech Recognition module, the sentence is transformed by the NLP module into the CI description:

(ask-if:OK #⟨SelectMessage #X107EG21⟩).

This CI description is passed to the DP module. A process flow for the CI type "ask-if:OK" was designed to ask a question like S31 if the user said an utterance like U30 in the situation when the user's task plan was unknown. If the user responds with a sentence fragment like U32 for S31, U32 is interpreted as the sentence "メッセージを移動したい (I want to move a message.)". That is, the process flow for "ask-if:OK" finishes processing, and then the process flow for "have-goal" starts processing.

5.5 Conclusion

In this section, the experimental user support system was shown. The main modules, the NLP and DP modules, of the support system were developed based on the analyses of real advisory dialogue data. The support system has the following features.

(1) The target of support is XMH with a visual interface, rather than with a command interface. In a visual interface, the commands are executed mainly by mouse operations.

(2) The support system recognizes the user's speech which was inputted by a microphone, and constructs a CI description which describes the user's communicative intention [6][7].

(3) An appropriate system response is generated using the response generation rules prepared for each CI type. The rules achieve the know-how which the consultant employed in order to effectively use the state of XMH and the event history of user XMH operations [13][14].

(4) The support system controls and manages dialogues with a user. Even if the user produces a subdialogue, the support system can restart the original dialogue after the subdialogue is concluded [15][16].

The support system can manage and control dialogues with a user, so that it can guide the user to his task goals even if the user produces a subdialogue or does misoperations.

We are considering evaluating the experimental user support system. First, what are evaluated and how they are evaluated should be discussed.

The CI recognition method developed in Section **3** deals only with a simple sentence and a complex sentence with at most one performative verb. This is due to the insufficient specifications for the utterance proposition construction stage. The next objective is to extend the stage so as to deal with a complex sentence with two or more performative verbs. A method should be developed to automatically remove or modify such linguistic phenomena as interjections, corrections, hesitations, paraphrases, and inversions.

Now, the modality information is extracted from a spoken sentence to determine the CI type of the sentence. Generally, however, uncertainty, regret, impatience, or other delicate attitudes can be extracted from a spoken sentence. A method for extracting these kinds of information and a method for effectively using them is also our next target.

The support system can provide only passive support. This means that the support system only responds to the user question. However, in the experiments for collecting the dialogues, active support [18] by a human consultant was observed. Such active support is another target of future work.

The support system has a multimedia interface which enables the speech input, keyboard input, and mouse input. A multimodal interface which enables an input by pointing gesture also becomes the next target.

Nagao et al. are doing experiments in which the mental barrier occurring in a dialogue with a machine is reduced by presenting facial displays [19]. This is a very important approach and has become one out of future work.

References

[1] Kumamoto, T., Ito, A., and Ebina, T.: "Design and development of a dialogue-based consultant system for supporting the use of electronic mail," *Journal of the CRL*, Vol. 42, No. 2, pp. 161-171 (1995).

[2] Kumamoto, T., Ito, A., and Ebina, T.: "An experimental dialogue-based consultant system for XMH users," *Proc. the 7th Annual Conference of JSAI*, 20-3, pp. 555-558 (1993) (in Japanese).

[3] Wilensky, R., Arens, Y., and Chin, D.: "Talking to UNIX in English: An overview of UC," *Communications of the ACM*, Vol. 127, No. 6, pp. 574-592 (1984).

[4] Uehara, T., Uehara, K., and Toyoda, J.: "Simulating user's consideration for inferring user's intention," *Technical Report of JSAI*, SIG-HICG-8902-2(10/4), pp. 11-20 (1989).

[5] Aoe, J., Maeda, A., Kujime, H., and Morimoto, K.: "Understanding of commands using natural language inputs — An aid to UNIX systems —," *Technical Report of IEICE*, NLC90-44, pp. 25-32, (1990).

[6] Kumamoto, T., Ito, A., and Ebina, T.: "Recognizing user communicative intention in a dialogue-based consultant system," *Trans. IEICE*, Vol. J77-D-II, No. 6, pp. 1114-1123 (1994) (in Japanese).

[7] Kumamoto, T., Ito, A., and Ebina, T.: "An analysis of Japanese sentences in spoken dialogue and its application to communicative intention recognition," *Proc. International Conference on Spoken Language Processing (ICSLP '94)*, Vol. 2, No. S17-19.1, pp. 943-946, Yokohama, Japan (1994).

[8] Masuoka, T.: "Modarithi no bunpou (Grammar of modality)," Kurosio Publishers, Tokyo (1991) (in Japanese).

[9] Kato, T., and Nakagawa, M.: "Logical form and Syntactic-Semantic Analysis for Natural Language Interface," *Proc. the 2nd Annual Conference of JSAI*, 9-2, pp. 479-482 (1988) (in Japanese).

[10] Den, Y., and Iida, H.: "An information based analysis of conversational sentences," *Technical Report of JSAI*, SIG-F/H/K-9101-12, pp. 111-120 (1991) (in Japanese).

[11] Ito, A., and Kumamoto, T.: "A semantics for a frame representation with circular concept definition," *Trans. IPSJ*, Vol. 34, No. 8, pp. 1731-1740 (in Japanese).

[12] Ito, A., Higuchi, H., Kumamoto, T., and Ebina, T.: "A knowledge representation for a dialogue system understanding natural language and command input in the unified context," *Proc. Japan-Australia Joint Symposium of Natural Language Processing*, pp. 95-105, Iizuka, Japan (1991).

[13] Ito, A., Ebina, T., and Kumamoto, T.: "A taxonomy of the user's questions and response strategy for dialogue based user support," *Trans. IEICE*, Vol. J77-D-II, No. 7, pp. 1319-1328 (1994) (in Japanese).

[14] Ito, A., Ebina, T., and Kumamoto, T.: "Dialogue-based user support for a visual interface software — A case study for XMH —," *Proc. Pacific Rim International Conference on Artificial Intelligence (PRICAI '92)*, 1, pp. 190-196, Seoul, Korea (1992).

[15] Kumamoto, T., and Ito, A.: "An analysis of user-consultant dialogue and its application to dialogue control in a dialogue-based consultant system," *Trans. IEICE*, Vol. J77-D-II, No. 8, pp. 1492-1501 (1994) (in Japanese).

[16] Kumamoto, T., and Ito, A.: "An analysis of user-consultant dialogues and its application to dialogue processing in a dialogue-based consultant system," *Systems and Computers in Japan*, Vol. 26, No. 3, pp. 103-114 (1995).

[17] Osaka Gas Information System Research Institute: "User's manual for speech recognition system DS200," release 1.01 (1992) (in Japanese).

[18] Koyama, S., and Yamamoto, Y.: "An intelligent help system," *Proc. the 8th Annual Conference of Japan Society for Software Science and Technology*, D2-1, pp. 157-160 (1991) (in Japanese).

[19] Nagao, K., and Takeuchi, A.: "A new modality for natural human-computer interaction: An experimental speech dialogue system with facial displays," *Technical Reports of JSAI*, SIG-SLUD-9204-2(2/5), pp. 9-16 (1992) (in Japanese).

6. Conclusions

This section is a conclusion of the present article, and contains the author's plans for future work.

6.1 Conclusions of the Present Article

A dialogue-based user support system is being developed that plays the role of a human consultant [1][2]. The support system provides support to XMH users through spoken dialogue. XMH, an X-window-based electronic mail handling program, has a visual interface operated mainly by using a mouse: all XMH command functions are implemented through mouse-controlled pull-down menus and command buttons. A user typically operates XMH through a mouse. When a user encounters trouble while performing a task, he can verbally ask the support system for help. The support system understands user speech, and gives responses appropriate to the state of XMH and the event history derived from user XMH operations.

When the support system talks with a user in order to assist him with the use of XMH, it requires at least two kinds of methods: one for understanding a user's speech, and one for controlling and managing dialogues with a user. In this article, each method was developed based on an analysis of real advisory dialogue data. For this purpose, first, data to be analyzed was collected. Then, each method was developed, and finally an experimental user support system was constructed using the methods. Conclusions for each section in the present article are given below.

Section **2** presented that an advisory dialogue database was constructed from the dialogue data collected in the XMH usage experiments [3][4]. This dialogue database contains advisory dialogues between 42 XMH users and a human consultant, and has the following four features.

(1) The dialogue database contains the spontaneous advisory dialogues, which are essentially goal-oriented.

(2) The advisory dialogues consist of user spoken sentences, user XMH operations, and consultant responses. A "spoken sentence" is one transcribed from an utterance, an "XMH operation" is either a mouse operation or a key operation, which is extracted from XMH as text data, and a "consultant response" is a Kanji-Kana mixed sentence with which the consultant responded in the XMH usage experiments, respectively.

(3) The user spoken sentences, as well as the user's speech, have the characteristics of spontaneous speech: linguistic phenomena such as interjections, repairs, hesitations, paraphrases, ellipses, and inversions.

(4) The dialogue database is available to people only for academic uses.

The dialogue database is a very effective source for developing new spoken-language processing methods, and was in fact used to develop the CI recognition method described in Section **3** and the dialogue processing method described in Section 4.

Section **3** uses user utterance data obtained from the dialogue database described in Section **2**, which consists of 475 spoken sentences; 298 sentences of the utterance data were analyzed. Based on the analysis, a description form of user communicative intention (CI) was defined from the viewpoint of the role of a user utterance in an advisory dialogue, and a CI recognition method was developed to construct a CI description from a spoken sentence [5][6].

A CI description is a semantic structure for representing CI, which a user utterance has, and consists of a CI type and an utterance proposition.

The focus of this section was a method for determining the CI type of a spoken sentence. The remaining 177 sentences of the utterance data were used to evaluate this method. According to the experiments done for evaluating the method, the method is accurate for all of the learned data (298 sentences) and for 89.3% of the unlearned data (177 sentences).

The CI recognition method mentioned above was implemented into the support system as the Natural Language Processing module.

In Section **4**, the dialogue data was obtained from the dialogue database described in Section **2**, and, based on an analysis of the dialogue data, a dialogue processing method was developed to control and manage advisory dialogues with an XMH user [7][8]. This method has the following features.

(1) Not only user utterances and consultant responses, but also user operations are managed as dialogue elements. Consequently, the dialogue processing method can detect user misoperation, inform the user of the fact, and suggest an appropriate recovery method.

(2) Advisory dialogues with a user are managed which may have one or more nested dialogues, where a nested dialogue is defined as a sequence of dialogue elements at the same depth.

(3) Advisory dialogues are controlled so that a user can achieve his task goal effectively.

The know-how of response generation that the human consultant employed to help the users was formalized as the response generation rules [9][10]. These rules were designed for each CI type, and were achieved by a "process flow" prepared for the CI type as an algorithm. The process flow is a mechanism for controlling a dialogue initiated by a user utterance with the CI type, which was developed in Section **4**.

The dialogue processing method mentioned above was implemented into the support system as the Dialogue Processing module.

In Section **5**, an experimental user support system [1][2] was constructed on the basis of the methods developed in Sections **3** and **4**.

This support system consists mainly of the Speech Recognition module, Natural Language Processing (NLP) module, Dialogue World model, XMH simulator, Dialogue Processing (DP) module, and Response Generation module. This article describes the NLP and DP modules in detail, because the modules correspond to the CI recognition method developed in Section **3** and the dialogue processing method developed in Section **4**, respectively.

Three examples of dialogues between the support system and a simulated user were presented in order to highlight the dialogue faculty of the support system. They were a typical advisory dialogue, a subdialogue interrupting the typical advisory dialogue, and an advisory dialogue containing a short sentence fragment.

6.2 The Author's Plans for Future Work

The support system described in this article provides only passive support to users. This means that the support system only responds to user speech. However, in the experiments made for collecting the advisory dialogues, active support [11] by a human consultant was also observed. Development of such active support is one goal of future work.

The support system has a multimedia interface which allows for speech input, keyboard input, and mouse input. However, in two-way dialogues between people, body language and gesture are also frequently used, and constitute an effective interaction method. Pointing gestures used with demonstratives are especially useful in an advisory environment. Consequently, a multimodal interface which accepts a pointing gesture will also be one focus of work.

Nagao et al. are doing experiments in which the mental barrier occurring in a dialogue with a machine is lowered by presenting facial displays [12]. This is a very important to the development of a user-friendly system, and has been targetted for future research.

Since the dialogue database described in Section 2 consists of text data, it would be very useful to also develop spoken-language processing methods. Similarly, speech data is an effective source for developing new speech processing methods. In the experiments, user speech was recorded on videotapes. Currently, we are considering the specifications of a speech database to be constructed at a later date.

The support system can only deal with a simple sentence and a complex sentence containing at most one performative verb. This is due to the insufficient specifications of the NLP module. The next objective is to extend the NLP module so that the support system can deal with a complex sentence containing two or more performative verbs.

Spontaneous speech often contains such linguistic phenomena as interjections, repairs, hesitations, paraphrases, ellipses, and inversions, and is abundant with demonstratives and anaphoric expressions. These are problems to be solved by the NLP module.

References

[1] Kumamoto, T., Ito, A., and Ebina, T.: "Design and development of a dialogue-based consultant system for supporting the use of electronic mail," *Journal of the CRL*, Vol. 42, No. 2, pp. 161-171 (1995).

[2] Kumamoto, T., Ito, A., and Ebina, T.: "An experimental dialogue-based consultant system for XMH users," *Proc. the 7th Annual Conference of JSAI*, 20-3, pp. 555-558 (1993) (in Japanese).

[3] Kumamoto, T., Ito, A., and Ebina, T.: "Design and construction of an advisory dialogue database," *IEICE Trans. on Information and Systems*, Vol. E78-D, No. 4, pp. 420-427 (1995).

[4] Kumamoto, T., Ito, A., and Ebina, T.: "Construction of a dialogue database from advisory dialogues between novice computer users and an expert consultant," *Proc. of International Workshop on Sharable Natural Language Resources (SNLR '94)*, pp. 139-146, Nara, Japan (1994).

[5] Kumamoto, T., Ito, A., and Ebina, T.: "Recognizing user communicative intention in a dialogue-based consultant system," *Trans. IEICE*, Vol. J77-D-II, No. 6, pp. 1114-1123 (1994) (in Japanese).

[6] Kumamoto, T., Ito, A., and Ebina, T.: "An analysis of Japanese sentences in spoken dialogue and its application to communicative intention recognition," *Proc. International Conference on Spoken Language Processing (ICSLP '94)*, Vol. 2, No. S17-19.1, pp. 943-946, Yokohama, Japan (1994).

[7] Kumamoto, T., and Ito, A.: "An analysis of user-consultant dialogue and its application to dialogue control in a dialogue-based consultant system," *Trans. IEICE*, Vol. J77-D-II, No. 8, pp. 1492-1501 (1994) (in Japanese).

[8] Kumamoto, T., and Ito, A.: "An analysis of user-consultant dialogues and its application to dialogue processing in a dialogue-based consultant system," *Systems and Computers in Japan*, Vol. 26, No. 3, pp. 103-114 (1995).

[9] Ito, A., Ebina, T., and Kumamoto, T.: "A taxonomy of the user's questions and response strategy for dialogue based user support," *Trans. IEICE*, Vol. J77-D-II, No. 7, pp. 1319-1328 (1994) (in Japanese).

[10] Ito, A., Ebina, T., and Kumamoto, T.: "Dialogue-based user support for a visual interface software — A case study for XMH —," *Proc. Pacific Rim International Conference on Artificial Intelligence (PRICAI '92)*, 1, pp. 190-196, Seoul, Korea (1992).

[11] Koyama, S., and Yamamoto, Y.: "An intelligent help system," *Proc. the 8th Annual Conference of Japan Society for Software Science and Technology*, D2-1, pp. 157-160 (1991) (in Japanese).

[12] Nagao, K., and Takeuchi, A.: "A new modality for natural human-computer interaction: An experimental speech dialogue system with facial displays," *Technical Reports of JSAI*, SIG-SLUD-9204-2(2/5), pp. 9-16 (1992) (in Japanese).

Acknowledgments

The author would like to express many thanks to all those who have provided kind and heartful support to his work and his research.

First of all, he must express special thanks to Professor Kazuo Toraichi, of the Institute of Information Sciences and Electronics at the University of Tsukuba, for his sincere supervision, hospitality, and encouragement.

Gratitude is due particularly to Dr. Akira Ito, a director of the Knowledge Systems section at the Kansai Advanced Research Center in the Communications Research Laboratory of the Ministry of Posts and Telecommunications, for his valuable comments, discussions and guidance provided to the author.

He also expresses sincere thanks to Professor Shuichi Itahashi, Professor Yukio Ishibashi, Professor Shojiro Nemoto, Professor Yoshihiko Ebihara, and Associate Professor Koichi Wada, of the Institute of Information Sciences and Electronics at the University of Tsukuba, for their valuable comments and discussions.

The author owes both Professor Kazuhiko Yamamoto in the Gifu University and Dr. Hiromitsu Yamada in the Electrotechnical Laboratory of MITI special thanks for their sincere supervision, hospitality, and encouragement.

He is also obliged to both Dr. Iwao Sekita of the Electrotechnical Laboratory of MITI and Associate Professor Masaru Kamada, of the Department of Computer and Information Sciences at Ibaraki University, for their guidance and encouragement throughout his research.

Finally, the author thanks Dr. Shinji Urabe, Director, and other members of the Kansai Advanced Research Center, in the Communications Research Laboratory of the Ministry of Posts and Telecommunications, for their kind consideration and hospitality.

The Author's Works

1. 査読あり (Refereed Papers)

[A]. 学会論文誌 (Papers in Journals)

A1 Tadahiko KUMAMOTO, Kazuo TORAICHI, Takahiko HORIUCHI, Kazuhiko YAMAMOTO and Hiromitsu YAMADA:
On Speeding Candidate Selection in Handprinted Chinese Character Recognition,
Pattern Recognition, Vol. 24, No. 8, pp. 793-799 (August 1991).

A2 伊藤昭, 熊本忠彦:
循環的な概念定義を含むフレーム表現の意味論,
情処学論, Vol. 34, No. 8, pp. 1731-1740 (August 1993).

A3 熊本忠彦, 伊藤昭, 海老名毅:
支援対話におけるユーザ発話意図の認識
— ユーザ発話文の解析に基づく統計的アプローチ —,
信学論, Vol. J77-D-II, No. 6, pp. 1114-1123 (June 1994).

A4 伊藤昭, 海老名毅, 熊本忠彦:
対話型計算機利用支援におけるユーザ質問の分類と支援回答戦略,
信学論, Vol. J77-D-II, No. 7, pp. 1319-1328 (July 1994).

A5 熊本忠彦, 伊藤昭:
支援対話の解析に基づく対話処理方式の提案,
信学論, Vol. J77-D-II, No. 8, pp. 1492-1501 (August 1994).

A6 Tadahiko KUMAMOTO, Akira ITO and Tsuyoshi EBINA:
Design and Construction of an Advisory Dialogue Database,
IEICE Trans. on Information and Systems, Vol. E78-D, No. 4, pp. 420-427 (April 1995).

A7 Kazuo TORAICHI, Tadahiko KUMAMOTO, Kazuhiko YAMAMOTO and Hiromitsu YAMADA:
Feature Analysis of Handprinted Chinese Characters,
Pattern Recognition Letters, Vol. 17, No. 7, pp. 795-800 (June 1996).

以上

[B]. 国際会議論文 (Conference Papers)

B1 Kazuo TORAICHI, Tadahiko KUMAMOTO, Iwao SEKITA, Kazuhiko YAMAMOTO and Hiromitsu YAMADA:
Feature Analysis of Handprinted Chinese Characters for a Recognition System,
Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '89), No. M6.5, pp. 1715-1718, Glasgow, Scotland (May 1989).

B2 Tadahiko KUMAMOTO, Kazuo TORAICHI, Kazuhiko YAMAMOTO and Hiromitsu YAMADA:
On Speeding Candidate Selection in Handprinted Chinese Character Recognition,
Proc. of IASTED International Symposium Machine Learning and Neural Networks (MLNN '90), No. 172-043, pp. 18-21, New York, U.S.A. (October 1990).

B3 Akira ITO, Hideyuki HIGUCHI, Tadahiko KUMAMOTO and Tsuyoshi EBINA:
A Knowledge Representation for a Dialogue System Understanding Natural Language and Command Input in the Unified Context,
Proc. of Japan-Australia Joint Symposium of Natural Language Processing (JAJSNLP '91), pp. 95-105, Iizuka, Japan (October 1991).

B4 Tadahiko KUMAMOTO and Akira ITO:
Recognizing User Communicative Intention in a Consultant System with a Natural Language Interface,
Proc. of IEEE International Workshop on Robot and Human Communication (ROMAN '92), pp. 311-316, Tokyo, Japan (September 1992).

B5 Akira ITO, Tsuyoshi EBINA and Tadahiko KUMAMOTO:
Dialogue-Based User Support for a Visual Interface Software
— Case Study for XMH —,
Proc. of Pacific Rim International Conference on Artificial Intelligence (PRICAI '92), No. 1, pp. 190-196, Seoul, Korea (September 1992).

B6 Tadahiko KUMAMOTO, Akira ITO and Tsuyoshi EBINA:
User Communicative Intention and the Generation of Helpful Responses
— A Consultant System for XMH Users —,
Proc. of Singapore International Conference on Intelligent Systems (SPICIS '92), pp. 543-548, Singapore (September 1992).

B7 Tadahiko KUMAMOTO, Akira ITO and Tsuyoshi EBINA:
Construction of a Dialogue Database from Advisory Dialogues between Novice Computer Users and an Expert Consultant,
Proc. of International Workshop on Sharable Natural Language Resources (SNLR '94), pp. 139-146, Nara, Japan (August 1994).

B8 Tadahiko KUMAMOTO, Akira ITO and Tsuyoshi EBINA:
An Analysis of Japanese Sentences in Spoken Dialogue and Its Application to Communicative Intention Recognition,
Proc. of International Conference on Spoken Language Processing (ICSLP '94), Vol. 2, No. S17-19.1, pp. 943-946, Yokohama, Japan (September 1994).

B9 Akira ITO, Tadahiko KUMAMOTO and Tsuyoshi EBINA:
Supporting Computer Users Through Dialogue,
Proc. of International Conference on Human-Computer Interaction (HCI '95), No. 2, pp. 65-70, Yokohama, Japan (July 1995).

B10 Tadahiko KUMAMOTO, Akira ITO and Tsuyoshi EBINA:
Improved Communicative Intention Recognition in Spoken Japanese Sentences by Extracting Propositional and Modality Information Separately,
Proc. of International Conference on Computer Processing of Oriental Languages (ICCPOL '95), pp. 68-75, Hawaii, U.S.A. (November 1995).

以下，採録につき，発表予定．

B? Tadahiko KUMAMOTO and Akira ITO:
Semantic Interpretation of a Japanese Complex Sentence in an Advisory Dialogue,
Proc. of International Conference on Spoken Language Processing (ICSLP '96), Philadelphia, U.S.A (October 1996).

以上

2．査読なし (Technical Reports)

[F]. 研究会報告 (Reports of Technical Group)

F1 関田巌, 熊本忠彦, 寅市和男, 森亮一, 山本和彦, 山田博三, 胡家忠:
手書き中国漢字データベースの作成と解析,
信学技報, CPSY87-8, pp. 9-16 (August 27, 1987).

F2 熊本忠彦, 伊藤昭:
対話型計算機利用支援システムにおけるユーザ発話意図の表現,
信学技報, HC91-46, pp. 33-40 (January 24, 1992).

F3 熊本忠彦, 伊藤昭:
ユーザ発話文の解析に基づく発話意図認識手法の提案,
信学技報, NLC93-10, pp. 73-80 (May 20, 1993).

F4 熊本忠彦, 伊藤昭:
対話型計算機利用支援システムのための対話制御方式,
人工知能学会研資, SIG-J-9301-2, pp. 9-16 (December 10, 1993).

F5 熊本忠彦, 伊藤昭, 海老名毅:
命題情報とモダリティ情報の個別抽出に基づく発話意図解析手法について,
人工知能学会研資, SIG-SLUD-9403-2, pp. 7-12 (February 10, 1995).

F6 熊本忠彦, 伊藤昭:
発話意図タイプ決定ルールの事例からの学習,
信学技報, NLC95-14, pp. 51-56 (July 20, 1995).

F7 熊本忠彦, 伊藤昭:
支援対話に現れる述語接続助詞「けど」の意味解析,
情処研報, 95-NL-110, pp. 7-12 (November 17, 1995).

以上

[G]. 全国大会講演論文 (National Convention Records)

G1 伊藤昭, 樋口英幸, 海老名毅, 熊本忠彦:
対話による計算機利用支援　（１）システムの構成と対話モデル,
情報処理学会第 42 回全国大会, No. 4C-7, pp. 3-72 – 3-73 (March 13, 1991).

G2 熊本忠彦, 伊藤昭:
対話による計算機利用支援　（３）知的支援のための自然言語処理,
情報処理学会第 42 回全国大会, No. 4C-9, pp. 3-76 – 3-77 (March 13, 1991).

G3 伊藤昭, 樋口英幸, 海老名毅, 熊本忠彦:
時間概念を持つフレーム表現の試み,
日本ソフトウエア科学会第 8 回大会論文集, No. B3-2, pp. 237-240 (1991).

G4 熊本忠彦, 伊藤昭:
自然言語質問応答システムのための発話意図の表現,
平成 3 年電気関係学会関西支部連合大会, No. G8-30, p. G257 (November 3, 1991).

G5 伊藤昭, 海老名毅, 熊本忠彦:
対話型計算機利用支援システムにおけるユーザ対話の解析,
人工知能学会第 6 回全国大会, No. 17-2, pp. 619-622 (June 1992).

G6 伊藤昭, 海老名毅, 熊本忠彦:
対話型計算機利用支援における対話戦略,
情報処理学会第 45 回全国大会, No. 3F-03, pp. 3-147 - 3-148 (March 1992).

G7 熊本忠彦, 伊藤昭, 海老名毅:
対話型計算機利用支援システムにおける音声理解について,
平成 4 年電気関係学会関西支部連合大会, No. G8-10, p. G268 (November 1992).

G8 伊藤昭, 熊本忠彦:
循環的な概念定義を含むフレーム表現の意味論,
日本ソフトウエア科学会第 10 回大会論文集, No. C2-2, pp. 273-276 (June 1993).

G9 熊本忠彦, 伊藤昭, 海老名毅:
電子メール処理プログラム (XMH) 利用支援システムの試作,
人工知能学会第 7 回全国大会, No. 20-3, pp. 555-558 (July 1993).

G10 熊本忠彦, 伊藤昭, 海老名毅:
命題情報とモダリティ情報の個別抽出に基づく文解析手法の提案,
情報処理学会第 49 回全国大会, No. 4G-7, pp. 167-168 (September 1994).

以上

[H]. 商業雑誌 (Comments in Commercial Journals)

H1 Tadahiko KUMAMOTO, Akira ITO and Tsuyoshi EBINA:
Recognizing User Communicative Intention in a Dialogue-Based Consultant System — A Statistical Approach Based on the Analysis of Spoken Japanese Sentences —,
Systems and Computers in Japan, Vol. 25, No. 14, pp. 91-105 (December 1994).

H2 Tadahiko KUMAMOTO and Akira ITO:
An Analysis of User-Consultant Dialogues and Its Application to Dialogue Processing in a Dialogue-Based Consultant System,
Systems and Computers in Japan, Vol. 26, No. 3, pp. 103-114 (March 1995).

以上

3．その他(Other)

[X]. 査読付き機関誌(Refereed Papers in CRL)

X1 Tadahiko KUMAMOTO, Akira ITO and Tsuyoshi EBINA:
Design and Development of a Dialogue-Based Consultant System for Supporting the Use of Electronic Mail,
Journal of the CRL, Vol. 42, No. 2, pp. 161-171 (July 1995).

X2 Tadahiko KUMAMOTO, Akira ITO and Tsuyoshi EBINA:
Communicative Intention Recognition in a Dialogue-Based Consultant System
— A Statistical Approach Based on the Analysis of Spoken Japanese Sentences —,
Journal of the CRL, Vol. 42, No. 2, pp. 181-198 (July 1995).

以上

[Y]. 査読なし機関誌(Technical Reports in CRL)

Y1 伊藤昭, 樋口英幸, 海老名毅, 熊本忠彦:
人と計算機の対話によるユーザ支援,
通信総合研究所第80回研究発表会予稿, pp. 37-46 (May 29, 1991).

Y2 熊本忠彦:
対話型計算機利用支援システム
— 対話で利用者を支援する —,
CRL ニュース, No. 194, pp. 1-3 (May 1992).

Y3 熊本忠彦, 伊藤昭, 海老名毅:
電子メール利用支援システム
— マルチメディア対話による利用者の支援 —,
通信総合研究所第87回研究発表会予稿, No. 4, pp. 36-43 (November 30, 1994).

以上