ICOT Technical Memorandum: TM-0094

TM-0094

Prototyping a Dialoging System with a Topic Management Function

Taizo Miyachi, Koichi Furukawa
(ICOT)
Hitomi Shimada, Shozo Kondo and Takashi Dasai
(Mitsubishi Electric Corp.)

February, 1985

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Prototyping a Dialoging System with a Topic Management Function

T. Miyachi+, H. Shimada++, S. Kondo++, T. Dasai++ and K. Furukawa+

+ *Institute for New Generation Computer Technology (ICOT)*
++ *Mitsubishi Electric Corp. Information Systems & Electronics Development Labo.*

ABSTRACT

Basic functions needed for cooperative dialogs between a user and a database system are outlined. Consideration is restricted to target-oriented type dialog and the database system does not use complicated strategy based on expert knowledge. The main characteristics of the dialog system (TODAY) are as follows: 1) Multiple sentences can be used in a single utterance, 2) pronouns, conjunctions, and ellipsis are allowed, 3) topics can be changed at any time, and 4) cooperative dialog functions can be supported by processing presumptions and utilizing relevant information.

1. INTRODUCTION

The computer systems of recent years provide capabilities far exceeding simple number crunching. They can perform advanced formula manipulation, such as database processing; some are even sophisticated enough to give expert advice. These advanced computer systems, however, will not be in wider use as human's "intelligent assistants" without functions which permits users to converse easily with machines. To achieve this, studies are required to clarify the basic capabilities and natural form of man-machine conversation [1,2,3,4,5,6,7,8,9,11,16].

In general, larger-capacity databases in the near future are expected to contain useful data and knowledge, plus the large volumes of data required by expert systems. We look forward to the time when we can build a conversational system that substitutes for a knowledgeable responder, which searches the object knowledge also looking up where and how to access target knowledge. As a first step, here at ICOT we are developing prototype modules called TODAY (Topic Change and Relevant Information Manipulating Dialog System) that seem necessary for a large-scale knowledge management system.

TODAY is being developed using Extended Self-contained Prolog (ESP), a logic programming language running on the ICOT-developed personal sequential inference machine (PSI). At the same time, we are examining what functions are required by a dialog system to converse with the user to provide data and knowledge in the database. The investigation mainly aim at interactive functions independent of knowledge in a particular application field, assuming that queries which use functions depending upon expert's special knowledge (such as a planning function) are handled by individual expert systems in the knowledge-base management system. As shown in Figure 1, the dialog system described in this report is responsible for message transfer and conversation.

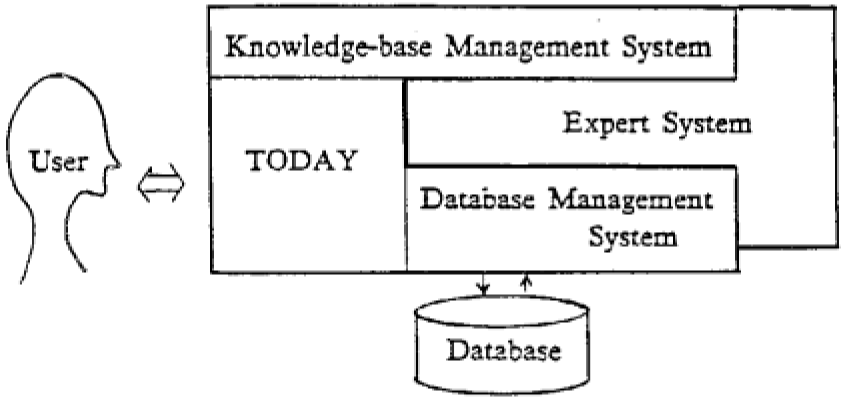

Figure 1 Relationship between Dialog System and Other Systems

2. Dialog Types and Processing Methods

In general, dialog systems can be roughly classified into two types depending on whether the dialogs handled by a particular system have any objective to be achieved or not:

(1) Fun-seeking types: Dialog systems which understand conversations undertaken just to have fun, with no other specific purpose.
(2) Target-oriented type: Dialog systems which understand conversations undertaken to achieve an objective, aiming to achieve that objective.

The dialog system discussed in this paper is of the target-oriented type. It aims to provide informaion required by the user. In a dialog for accomplishing a certain target, intention and requirements must be correctly transferred between the two parties concerned. This requires expressions used in the dialog to be simple, precise, and easy to use and understand. Our dialog system is designed to understand dialogs in a pseudo natural language with the following characteristics:

(1) It consists of restricted Japanese.
(2) It allows each message to consist of multiple sentences.
(3) It allows a restricted use of pronouns, conjunctions, and ellipses.
(4) It permits simple arithmetic expressions.

2.1 Search Request Consisting of Multiple Sentences

Writing and understanding sentences which include embedded sentences or complex dependencies is not easy in many cases. On the other hand, a dialog system capable of understanding a multiple sentence-based search request can eliminate processing in an unnatural manner in which users must query the system with a single sentence packed with many search conditions. In TODAY, the user can not only describe search items and their requirements in any order, but also use pronouns and conjunctions to omit redundancy. This permits requests to be concisely expressed. Furthermore, misunderstanding of complex expressions can be avoided.

(Ex. 1) Search requests consisting of a single sentence and multiple sentences

[Single sentence-based request]

プログラム言語＝ COBOLで 終了日＞74年のprojectID は？

(Find the project IDs which satisfies "programming language = COBOL" and "expiration date > 1974".)

[Multiple sentence-based request]

projectID について 知りたい	(I would like to know about project IDs.)
プログラム言語＝COBOL です	(Programming language = COBOL.)
さらに 終了日＞74年の ものは？	(Expiration data > 1974.)

Both requests are converted into the same internal expression called the pre-interim format.

2.2 Message Entry and Processing Methods

Input Japanese messages at present are assumed to be written in the Roman alphabet with a space between words.This is due to the incomplete Kanji (Chinese character) handling capability of the current version of PSI. This approach helps make messages more legible and the system more easily expandible. At the end of each sentence, the user is required to enter a mark called a full-stop mark followed by the Return key.

Dialogs are handled by six modules: sentence analysis, context handling, dialog management, command generation, response management, and window management. Input messages are first converted by the sentence analysis module into the pre-interim format. The subsequent two modules, context handling and dialog management, then add omitted and related information to the output of the sentence analysis module to generate the message in the interim format. The command generation module generates appropriate command, according to the interim format, which are then executed by the response management module (See Figure 2.1).

3. Context Processing

The use of pronouns and conjunctions to implement ellipses is important in conversation, because they can simplify sentences and help conversation flow smoothly. The context handling module uses any pronoun and conjunction information in the pre-interim format to recognize the existence of pronouns and conjunctions. Then it uses context and directory information to determine the referents of the pronouns and understand the meaning of the conjunctions, so that any ellipsis can be identified and added to the pre-interim format. It also initiates the functions which store context information and store and confirm dialog pairs.

Since many pronouns and conjunctions perform similar functions, we have minimized the range of pronouns and conjunctions to be used in the dialog system, so long as an adequate level of expression ability can be assured. Note that the system has been structured to permit easy addition of new pronouns and conjunctions.

(1) Pronoun Processing

Pronouns available in the system can be classified into two types as to whether a particular pronoun indicates an object or condition:

1) Pronouns identified in the pre-interim format as objects.
2) Pronouns identified in the pre-interim format as conditions.

An example of the former is "mono (one)"; the latter, "sore (it)".

(Ex. 2) では、プログラム言語＝FORTRAN のものは？
(Find the ones satisfying "programming language = FORTRAN.")

(Ex. 3) また、それがPL/1では？
(What if it is written in PL/1?)

Pronoun processing is conducted in the following manner: The context handling module recognizes the existence of any pronoun using the pronoun information in the pre-interim format sentence sent from the sentence analysis module. It then performs anaphora on the context information obtained through the dialog management module. Then it uses the directory information to validate the anaphora-proposed value. As a result, the pronoun information in the pre-interim format sentence is replaced with an appropriate specific value.

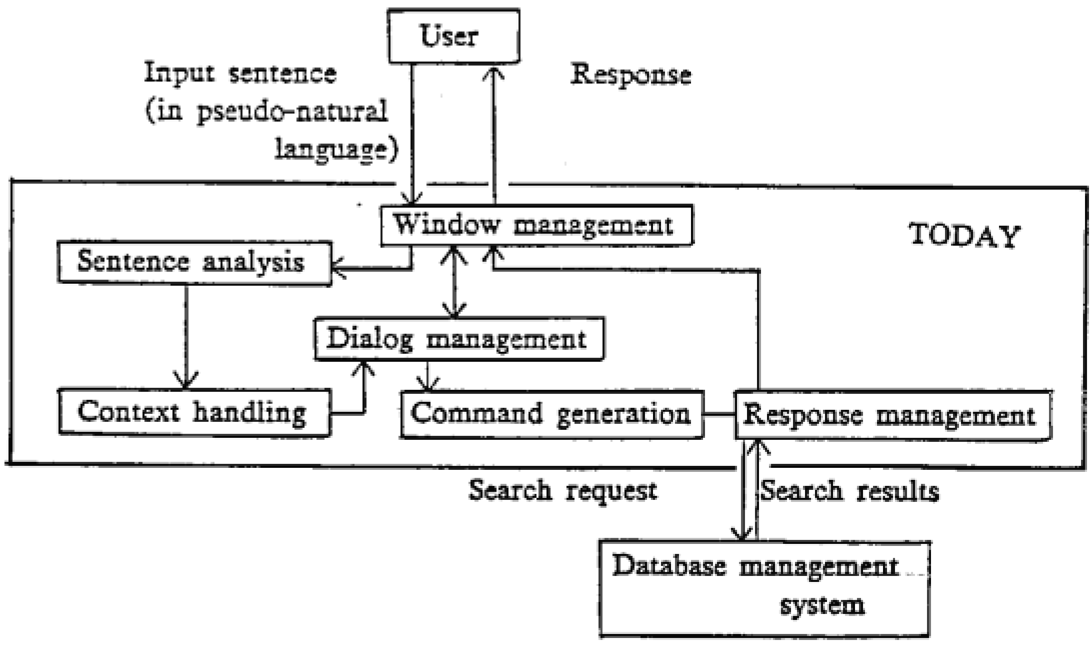

Figure 2.1 Process Flow in Dialog System

When a search request consisting of multiple sentences includes any pronoun, the sentence analysis module handles the pronoun using interim analysis results in the pre-pre-interim format and the directory information. If no appropriate specific value is found, the module informs the dialog management module that the system must ask the user to supply necessary information.

(2) Ellipsis Processing

Ellipses can be implemented using pronouns and conjunctions. All pronouns and the following six conjunctions can be used: "tokorode (by the way)," "sonotokino (under the same conditions)," "sarani (furthermore)," "sononakade (of these)," "dewa (well)," and "mata (also)."

The context handling module uses any conjunction information in a pre-interim-format sentence to identify the presence of ellipsis, extracts the omitted information from the context information of the interim format of previously handled sentences, and finally adds it to the pre-interim format sentence.

(Ex. 4) そのときのシステム設計書の規模は？
(Find the size of the system design document under the same conditions.)
(Ex. 5) また、プログラム言語＝FORTRAN では？
(Also find the ones satisfying "programming language = FORTRAN.")

4. Dialog Management

In target-oriented dialog the following three points are crucial.
1) To produce high-level results.
2) To rapidly advance conversation.
3) To cooperatively advance conversation.

1) means that a dialog must be able to achieve all its targets at a satisfactory level. It is indispensable for this condition that investigations be carried out not from only one point of view but from various point of view. The change of viewpoints causes a topic change. The use of a function for topic changes makes it easy to realize 2). The function for 3) is useful when a user does not know the information indispensable for attaining the objects and the user must search for the information. A topic change is triggered when the user starts the search. Therefore, the functions for managing topic change are indispensable.

The dialog management module in TODAY can store context information, check whether each query from the user has been given the corresponding response from the system, recognize the topic of a dialog, manage any topic change, and perform other processing to determine pronouns, handle ellipses, cooperatively respond to the user, and perform other operations. All of this permits conversations between the user and machine to progress in a natural manner. The dialog management module has three functions: context management, topic management (See Figure 4.1).

In this section we investigate the following six items:

(1) Tracing the flow in a dialog
(2) Leadership

(3) Checking consistency in a dialog
(4) Context management
(5) Topic management
(6) Dialog pair management

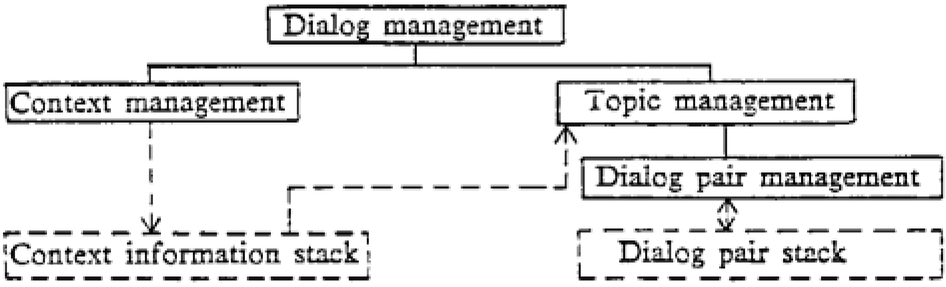

Figure 4.1 Structure of Dialog Management Module

4.1 Tracing Flow in a Dialog

Basically any dialog can be considered to consist of "dialog pairs" [16] each made up of a message from the user and the corresponding response from the system. In fun-seeking dialogs, however, dialog pairs may fail to form, because conversation progresses with more stress placed on the interesting aspects of the topic. But target-oriented dialogs always consist of dialog pairs, because cooperative conversation is performed to obtain better results.

Dialog pairs can be classified according to their relationships: greeting versus greeting, question versus answer, suggestion versus selection, order versus fulfilment, suggestion versus recognition, and so on. These relationship types are stored in the "dialog pair dictionary" in the system. In general, dialogs consisting of dialog pairs often have a "nested structure" in which a dialog pair contains another dialog pair [8]. To identify this structure, it is important in target-oriented dialogs to trace the flow in a dialog using dialog pairs.

Also a dialog performed to obtain information consists, in many cases, of dialog pairs connected with each other to express some request, except the case when the dialog is undertaken to state search conditions.

4.2 Leadership

To efficiently perform a target-oriented dialog, the person who knows the best method to achieve the objective of each scene should assume leadership of the conversation. Since different people are likely to have different opinions, however, the criterion to determine "the best" differs from person to person and thus it is not uncommon that the dialog fails to complete. Also humans may suddenly change their minds or have inconsistent criteria.

Yet, at present, man-machine conversation should basically be performed giving a higher priority to humans; the user should be able to assume leadership of the conversation whenever he wants. The machine should work following a plot shown by the user and have the leadership only when individual detailed targets are to be reached.

4.3 Dialog Consistency

Since a dialog consists of multiple dialog pairs connected with each other, different types of consistency can be found in different parts of the dialog. Consistency types found in a dialog can be divided into these four types:

(1) Dialog pair consistency
(2) Topic consistency
(3) Target consistency
(4) Other consistency

Each portion of a dialog seems to have at least one of these consistency criteria. A dialog pair has consistency when it "closes," or a message from the user has the corresponding response from the system as described in 4.1. The second type depends upon the content of a topic; a sentence in a dialog is related to another sentence in terms of the focused object or information relevant to the object. A target has consistency if any established target is ultimately accomplished. The fourth type indicates that dialog sentences are related to each other in terms of association (excluding (1) to (3)) or that there is no relation between them.

For systems, such as our system at ICOT, handling dialogs performed to achieve a certain goal, the third consistency criterion is necessary. Also consistency in a dialog pair is necessary to make the conversation more efficient and accomplish the target faster. A cooperative conversation should be conducted under this condition. The second consistency criterion, however, is not always required to provide higher-level results. For example, when not enough information is available for accomplishing an interim target, the topic of the conversation may be replaced with another apparently unrelated topic to obtain necessary information. Thus the first and third consistency criterion are the most important in this system.

4.4 Context Management

The context management function manages, as context information, 1. the target of a dialog, 2. the requirements of the target, and 3. the interim format generated from a given dialog pair, using separate "context-information stacks." It stores context information, derived from the pronoun and ellipsis processing performed by the context processing module, in context-information stacks. When the context processing module fails to find information for determining the referent of any pronoun and ellipsis processing performed by the context processing module, the context management function asks the user to supply the necessary information. It also provides context information to the command generation module when it generates commands and to the response management module when it generates a cooperative response.

As with dialog pair stacks discussed later in this paper, the context-information stacks contain dialog history and are equivalent to human short-term memory. Therefore, they have a finite depth, a fixed length of n in our system.

4.5 Topic Management (Recognition of topic change)

In general, dialogs handle multiple events or are performed from multiple perspectives to obtain higher-level results. This leads to changes in the topic of a dialog. Also when the user who plays the major role in a conversation has insufficient knowledge and cannot adequately understand the contents of the conversation, another user, or the dialog system, must temporarily change the topic of the dialog to provide the user with necessary knowledge. Thus it is important to recognize changes in the topic of conversation to correctly understand the contents of the dialog and to permit simplified expressions using pronouns and ellipses [7]. The topic management function is accessed by the context handling and command generation modules to store and check dialog pairs as well as to detect and handle any topic changes. The topic of a dialog changes:

(1) When the leader of the conversation changes the objective, or
(2) When the leader of the conversation is replaced with a new leader, who then sets a new objective.

In general, a dialog does not proceed smoothly unless the speaker and all the listeners correctly recognize topic changes. When the leadership of the conversation moves from one person to another, only the new leader or some of the listeners may detect a topic change; to avoid this situation, a change in topic is often explicitly mentioned.

In our dialog system, the topic management function searches the dialog pair dictionary for the expected response type according to a pre-interim-format sentence sent from other modules, and passes it to the dialog pair management function. When the dialog pair management function returns a response, this is used together with information on the object and conjunction information in the pre-interim-format sentence to judge whether the topic has changed. The following criterion is used in the judgement:

<Criterion to determine a topic change>

A topic change occurs when the message given by the user in response to a question from the system fails to form a dialog pair and aims at a different object. (Note that the object here is recognized in pre-interim-format sentences.)

4.6 Dialog Pair Management

The dialog pair management function stores and checks a dialog pair in a dialog pair stack according to a command from the topic management module. When a change in conversation topic is detected, it uses another dialog pair stack to store the message causing the topic change. When the new stack has a closed dialog pair, the dialog pair management function asks the user to confirm that the new topic has been exhausted before the original topic is recovered. (See Figure 4.2).

As with the context-information stacks, the dialog pair stacks have a fixed depth of n.

(Ex. 6) Moving to a new topic and then returning to the original topic

(User 1) 開発環境がバッチでプログラム言語がＣＯＢＯＬのプロジェクトＩＤは？

(Find the project IDs which satisfies the conditions: the development environment is batch processing and the programming language is COBOL.)

(System 1)

Project ID	Development environment	Programming language	Control Program
PA05	Batch	COBOL	CICS
PT33	Batch	COBOL	TOTAL
PC57	Batch	COBOL	DI/1

(User 2) ではプログラム言語がＦＯＲＴＲＡＮのものは？
(What if the programming language is FORTRAN?)
(System 2) その他の検索条件はそのままですか？
(Are the other search conditions unchanged?)
(User 3) 開発環境のレンジについて知りたい
(I would like to know the types of development environments.)
(System 3) 開発環境にはバッチとオンがあります
その他に質問はありますか？
(Development environment is either batch or online processing.
Are there any other questions?)
(User 4) いいえ (No.)
(System 4) では先程の話に戻って．
その他の検索条件はそのままですか？
(Let us return to the original topic.
Are the other search conditions unchanged?)

Thus, the dialog management function consisting of context management, topic management, and dialog-pair management can realize a cooperative dialog by handling multiple topics and providing the user with necessary knowledge in the course of the conversation.

5. Search Function for Cooperative Discourse

The causes of interference in a cooperative discourse are roughly classified into three categories.

1) Users' mistakes
  a) Misunderstanding of the premise of the object.
  b) Mistakes in input operations.
2) Misunderstanding of the discourse
  a) Misunderstanding about the number of responses.
    a-1) NIL response (including null search).
    a-2) Many answers.
  b) Misunderstanding of the content
    b-1) Mis-recognition of the target.
    b-2) Specification of information near the target (including a vague target)
3) Lack of effective capacity for predicting requests

1) refers to the user's misunderstanding of the target or mistakes in input operations. 2) refers to misunderstandings caused by insufficient understanding of the conversation or lack of expressive power. 3) means that effective predicting power is necessary.

To remove these type of interference the search function of a database must be able to:

(1) Retrieve data which satisfies the search conditions.
(2) Determine the efficacy of a search.
(3) Explain the search results.
(4) Cooperatively respond to messages from the user.
(5) Make use of the previous search results (this is called a narrowing capability).

The first capability is the same as the search function of the relational database; its optimizing technique was reported in [9]. Thus far our system has yet to be optimized. For features (2) to (4), presumption processing plays a major role [5,6]. The application of relevant information is effective in implementing (3) and (4). Our system is designed to provide these capabilities as well as the function for handling presumptions. Features (2) to (5) are described below.

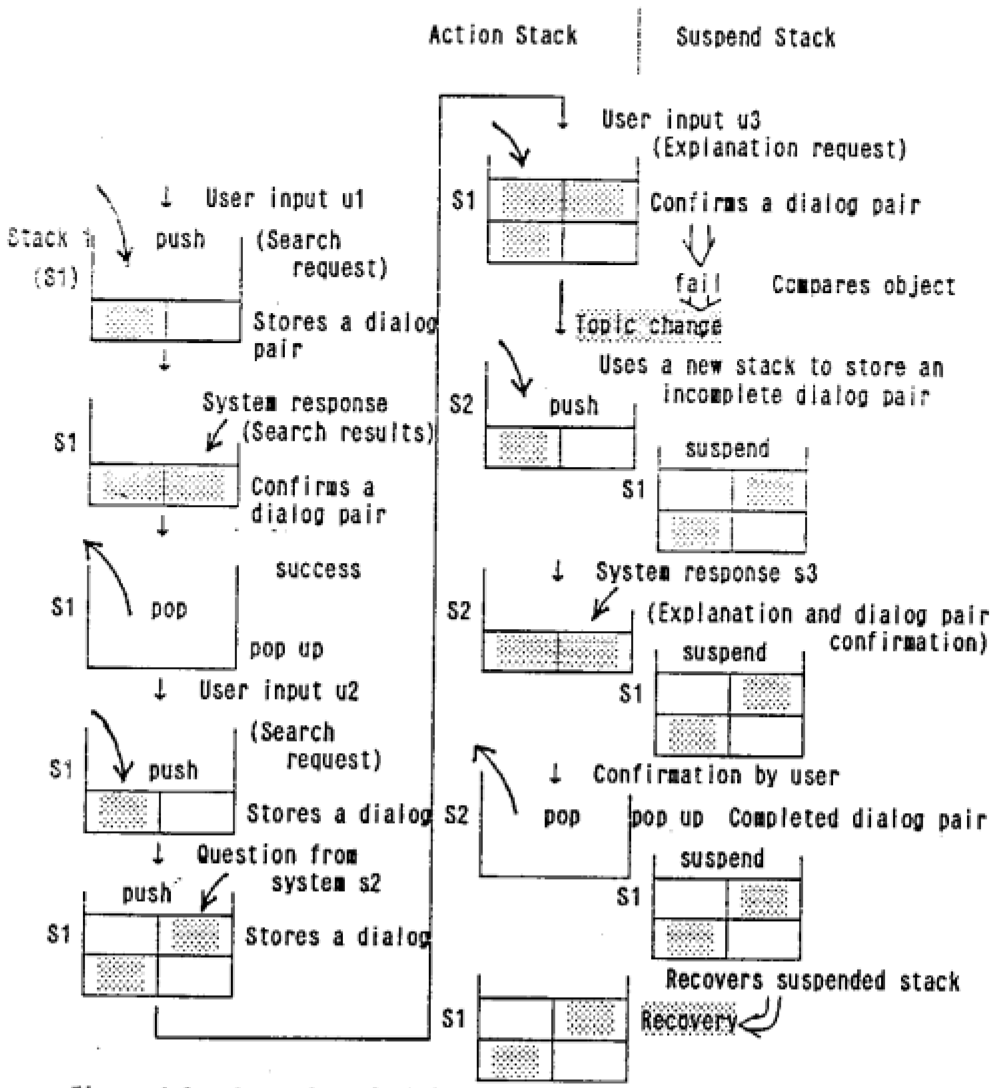

Figure 4.2 Operation of Dialog-Pair Stacks

5.1 Handling Presumptions

Our system detects a failed database search due to a presumption, finds the cause of the failure, and reports it to the user. Presumption-caused failures can be divided into two types:

(a) Those caused by extensional presumptions.

This type occurs according to the current values in the database.

(b) Those caused by intensional presumption.

This type occurs according to a basic structure in the database.

When a type (a) failure occurs, the so-called NIL response is returned. Causes for type (b) failures are divided into the following four:

1. Question concerning an undefined relation.
2. Attribute values having different data types.
3. Misunderstanding of characteristics of a relation.
4. Access to data which cannot maintain currency.

The system is capable of handling (a) and 1 and 2 in (b). This capability can be achieved with a function which recognizes the object of a search and checks whether it exists in the database. Attribute values have different data types when the user supplies: (1) An incorrect search object, or (2) An incorrect search condition.

The system at present does not handle 3 and 4 in (b), because these failures are more likely to occur for domain-specific reasons than 1 and 2 in (b).

5.2 Checking the Efficacy of a Search

Due to the user's insufficient knowledge or misunderstanding, search requests issued from the user do not always provide all the necessary prerequisites. This makes it necessary for the system to have a function for checking the efficacy of a given search request and analyzing faults.

Our system implements this function using the directory information in the database. This function also has the ability to handle failed searches caused by intensional performed, this system informs the user of this fact and prompts him to re-enter the correctly specified search condition.
incorrectly specified search condition.

(Ex. 7) Checking efficacy by extensional presumption.

(User) 開発環境＝ 2でプログラム言語＝COBOL のプロジェクトIDは？
(Find the project IDs which satisfy "development environment = 2 "and "Programming language = COBOL.")

(System) 開発環境の値は数値ではありません。
開発環境を入力しなおして下さい。
(The development environment value is not numerical. Enter the development environment again.)

5.3 Explaining Search Results

If search failed due to an extensional presumption (see 5.1 (a)), and , as a result, a "NIL response" is returned, the system examines the cause of the NIL response by comparing the search condition values and data values and using the relationship between data items. It then explains the determined cause to the user.

Even if a search is successful, the user may get an unexpected result. The system is equipped with an explanation function that displays the number of solusions, when it generates a large number of solutions which cannot all be displayed at one time. It then waits for the user to enter an instruction. It also has a contents-related explanation function which displays not only solutions but also the search conditions to implicitly ask the user for confirmation (see Example 8, 10).

5.4 Cooperative Response

When the user encounters an unexpected condition, the system can cooperatively respond to the user's messages [9,11] so that the subsequent dialog between user and system can be performed to achieve the dialog target. The system can cooperatively respond by:

(a) Correcting errors,
(b) Transforming or summarizing the solution, or
(c) Coopelatively advancing the conversation

Our system can be considered to indirectly implement the correcting errors, when a search fails due to a presumption, e.g. by displaying the cause of the error (the cause of a NIL responses or an impossible search). Since (b) mainly consists of database-dependent processing, the system is designed to handle units and add digit separators to implement (b) at a rudimentary level.

Adding relevant information:

Adding relevant information is effective for (c). The reasons for constructing efficient relevant information are four-fold:

1) The content of speech is restricted by the intention of the speaker who has a purpose or plans.
2) There are always experimental facts necessary for understanding the meaning behind the sentences or for understanding plans to attain the object [17].
3) To give an unusual meaning to a word, preparatory expressions must appear in the context several times.
4) Relevant information can transfer the objective information with less restrictions caused by the topics, because it covers various aspects of the object.

There are three levels according to the functions used to effect (c).

1) Offer of simple relevant information or background information.
2) Offer of plans for established processes as a system's belief.
3) Offer of subplans for a target near at hand.

The information given by 3) is efficient for building a total plan for a object.

TODAY has a rule-base using the context information and the dictionary information in the database to determine the necessary relevant information. TODAY searches the necessary relevant information and adds it to the search results.

(Ex. 8) Explanation given when an unexpected result (NIL response) is obtained due to a presumption.

(User) 終了日＞ 800年で プログラム言語＝COBOL のプロジェクトIDは？
(Find the project IDs which satisfies "expiration date > 800" and "programming language = COBOL.")

(System) 終了日は 1970年＜終了日≦1984年の範囲でなければなりません。
「終了日＞ 800年」と入力されたので該当する解がありません。
(The expiration date must be in the range: 1970 < expiration date < 1984. No solution can be found, because "expiration date > 800" was entered.)

(Ex. 9) Addition of relevant information.

(Ex. 9.1) See the system response (System 1) in the example in 4.6. ("control program" was added as relevant information.)

(Ex. 9.2) Relevant information consisting of multiple different data items

(User) マンナンバー＞8100で給料＞25万円の従業員は？
(Find the employees who satisfy "salary > 250,000" and "employee number > 8100.")
(System) Search results

Employee name	Employee number	Salary	Department	Phone number
Sato, H.	8275	260,000	Sales	2970
Suzuki, J.	8301	270,000	Accounting	2835

(Solutions) (Relevant information)

(Ex. 9.3) Offer of plans for established processes
(When the system's plan is as same as user's.)

U: 風邪薬を 下さい (Do you recommend any particular cold medicine ?)
S: どんな症状ですか？ (What are your symptoms ?)
U: 熱が でてるのです。 (I have some fever ?)
S: この薬は どうですか？ (I think this may work.)

Medicine name	Name of illness	Symptoms	Warning
Benze	Cold	Fever	Pirin
Rurum	Cold	Fever, ~~Cough~~	Non-pirin
Sutone	Cold	Fever, Stomachache	Non-pirin

U: 3番目のを 下さい (I will take the third one.)

S: はい、2500円です (2,500 yen, please.)

Ex. 9.3 shows what happens when the system's plan is the same as user's. The attributes check of a commercial item follows the check of functions in the system's plan. If a user omits some checks, the system adds the relevant information to press the user to pay attention the information omitted (e.g. price). Here, the lack of condition causes the system's first response.

(Ex. 9.4) Offer of subplans for a target near at hand.

S: お酒は 何に しますか (What will you have for drink ?)

Kinds	Name	Production	Price	Amount	Recommended appetizer
Wine	Beaujolais	French	3500	720	Fresh oysters
Beer	Korin	Japan	1000	633	Cheese & cracker

U: ワインを 下さい (A bottle of wine please.)
S: オードブルは？ (Would you like some appetizers ?)
U: フレッシュ・オイスタを 下さい (I'd like fresh oysters.)

In Ex. 9.4, at first, the user determines the kind of drink he wants to drink in a restaurant. The system kindly advises the user of appetizers suitable for the drink with relevant information.

5.5 Narrowing Function

In a conversation between man and machine, if the machine can first find an abstract object and later search the target of the conversation at high speed by considering the difference between the object and the target, the user need not examine the conditions of the conversation in detail. The narrowing function provides a way to achieve this.

This function searches efficiently, because it permits the results of the previous search to be used in the current search if the conditions of the current search are stricter than those for the previous search. In other words, if the current conditions consist of the previous ones plus new conditions. Our system equipped with this function initiates it either whenever a certain conjunctions "sononakade (of these)," "sonotokino (under the same conditions)," or "sarani (furthermore)" is encountered in a message from the user, or according to the relative strength of search conditions.

Many solutions: when a search result consists of more than n solutions, it is called "many solutions" and "n" is called a "solution-determining number" or simply "determining number."

The narrowing function is often required when many solutions are obtained during a dialog. The relevant information displayed can be used as attributes which determine the narrowing conditions.

(Ex. 10) Narrowing function (Example of processing many solutions obtained during a dialog.)
The solution-determining number n is assumed to be 20.

(User) プロジェクトIDについて　知りたい
開発環境が　バッチです
プログラム言語が　COBOL です
(Find the project IDs which satisfies the following conditions: The development environment is batch processing and the programming language is COBOL.)

(System) 20個以上の解が存在しますが、どうしますか？
次の３つの内から番号で選択して下さい。
(20 or more solutions can be obtained. Specify one of the following choices by number:)

1. no display.
2. Display some of the solutions (the number of solutions to be displayed must be supplied).
3. Display all the solutions.

(User) 1

(System) 105個の解が存在しました。
(105 solutions have been found.)

(User) そのなかで、システム要件書の規模>250 のものは？
(Find the ones whose system specification document size exceeds 250 pages.)

(System) 4 個の解が存在しました。
(Four solutions have been found.)

Development environment = batch, programming language = COBOL

Project ID	System specification document size (in pages)	System design document size (in pages)	Control program
PA07	300	620	CICS
PC87	255	354	TOTAL
PS11	350	765	TOTAL
PT88	340	450	DL/1

Note: The system design document, specification document sizes, and control program are added as relevant information.

In a dialog, the concept of many solutions can be considered to play the following role: At an abstract level, humans can choose a very wide area as the object of conversation. When the object area becomes more specific, however, they can handle a considerably smaller volume of information as the many solutions concept shows.

6. Conclusion

This paper examined the basic dialog functions required by advanced intelligent computer systems to converse with humans: Dialogs conducted between database systems and humans were studied. We reported on a dialog system that was experimentally developed here at ICOT and investigated under limited conditions eliminating the use of complicated knowledge, such as a planning function requiring special expert knowledge in a particular application area.

Our experimental dialog system permits Japanese in a limited form and multiple sentence-based messages as well as pronouns, conjunctions, and ellipses to be used in a dialog. The leadership of conversation is in most cases given to the human user. The system cannot take the leadership except for a single scene in which it is to achieve individual detailed subtargets according to the plot specified by the user. This approach permits considerably less restrictions to be placed on conversation.

For the dialog management function, the heart of the system, we examined the topic, context, and dialog pair management functions and their relationship. By means of this system, we tried to implement the abilities to cooperatively advance a dialog by handling multiple topics and/or providing the user with necessary knowledge in the course of the dialog. We developed the dialog pair management function so as to recognize when topic of conversation has been replaced with a new topic and to recover the old topic when the new topic is exhausted. We also studied ways to maintain the flow of a dialog. As a result, it was found that management of dialog pairs is an important basic function in overall management of target-oriented dialogs, and that the capabilities for maintaining conversation flow, recognize topic changes, and recover the previous topic work very effectively to implement flexible dialogs.

For the search function, we mainly investigated the facilities for determining the efficacy of a search, explain the search results, and cooperatively respond to the user. In database search, these facilities are closely related to the function that handles presumption and relevant information. For this reason, we examined the role of the presumption-processing function and the relevant information utilization function in realizing these functions. In addition, the narrowing capability was studied as a way of recognizing changes in the object area of a dialog.

In the future, we plan to implement a more cooperative dialog function capable of processing presumptions that strongly depends upon a particular application field, as well as a function to summarize the search results when the target area of a dialog changes.

Acknowledgements

We would like to thank Dr. Kazuhiro Fuchi, director of the ICOT Research Center, for providing us with the opportunity for this research. We would also like to extend our thanks to Messrs. Susumu Kunifuji, Hajime Kitakami, and Hideki Hirakawa, researchers at the ICOT Research Center, for contributing to the discussions.

References

1. Allen, J.F. and Perraut, C.R., "Analyzing Intention in utterances," AI15, 143-178, 1980.
2. Bobrow, D.G. et al., "GUS, A Frame-Driven Dialog system," AI, Spring, pp. 155-173, 1977.
3. Chikayama, T., "ESP Reference Manual," ICOT Technical Report (TR044), 1984.
4. Codd, E.F., "Rendezvous Version 1 : An Experimental English-Language Query Formulations System for Causal Users of Relational Data Bases," RJ2144(28407) Computer Science Research Report, IBM Research Lab. San Jose, 1978.
5. Grosz, B.J., "The Representation and Use of Focus in a System for Understanding Dialogs," Proc. IJCAI-77, pp. 67-76, 1977.
6. Harris, L.R., "User-Oriented Data Base Query with ROBOT Natural Language Query System," 3rd. International Conf. on VLDB, pp. 303-311, 1977.
7. Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D. and Slocum, J., "Developing a Natural Language Interface to Complex Data," ACM TODS, Vol. 3, No. 2, pp. 105-147, June, 1978.
8. Joshi, A.K., "Mutual Beliefs in Question Answering Systems," in N. Smith (editor), Mutual Belief, Academic Press, New York, 1982.
9. Kaplan, S.J., "Cooperative Responses from a Portable Natural Language Database Query System," Computational Models of Discourse, The MIT Press, pp. 107-166, 1983.
10. Nilsson, N.J., "Principles of Artificial Intelligence," Toiga Publishing Co., 1980. 6.
11. Nishida, T., "Discourse Model on a computer (survey)," Meeting for Cognitive Science of Discource, 1984.
12. Schank, R.C. and Abelson, R.P.,"Scripts, Plans, Goals, and Understanding," Hillsdale, N.J., Laurence Eribaum, 1977.
13. Suzuki, K. et al., "Applications and Utilizations of Knowledge for a Question Answering System in japanese," Institute of Electronics and Communication Engineers of Japan, AL82-70, 1982.
14. Waltz, D.L., "Natural Language Access to a Large Data Bases," Computers and People, pp. 19-26, April, 1976.
15. Warren, D.H., "Effective Processing of Interactive Relational Database Queries Expressed in Logic," Proc. of VLDB, pp. 272-281, 1981.
16. Yamanashi, M., "Basic Profiles in understanding discourses," Meeting for Cognitive Science of Discource, 1984.
17. Yasui, M., "Meanings behind sentences," Kenkyusha Press, 1978.