

# Spoken Dialog System for Database Access on Internet

Takuya NISHIMOTO, Yutaka KOBAYASHI, Yasuhisa NIIMI

Kyoto Institute of Technology  
Matsugasaki, Sakyo-ku, Kyoto, 606 Japan  
{nishi,koba,niimi}@dj.kit.ac.jp  
<http://www-vox.dj.kit.ac.jp>

## Abstract

In current spoken dialog systems, the design of user interfaces strongly depends on tasks and domains. We intend to develop a general purpose spoken dialog system. It extends existing Graphical User Interface (GUI) design used in on-line database services on the World Wide Web (WWW), so that the design of user interface does not require special knowledge about spoken dialogs. Web documents are written in Hyper Text Markup Language (HTML) and Common Gateway Interface (CGI), which naturally guide some dialog interactions as intended by the authors and by the browsers. In other words, each document has its own dialog control strategy in itself. Making use of such strategies enables task-oriented spoken dialogs, provided a single set of task-independent client modules. The dialog management information is extracted from mark-up tags, accordingly free from natural languages of text. Therefore, one could even provide a cross-language feature to the system by combining machine translation systems.

## Introduction

A speech interface system is regarded as a very natural way for unskilled people to use computers. If a system is carefully designed to be a good combination of conventional GUI and speech interface, it also increases the productivity.

Many standards of GUI environment are proposed, but Motif, MS-Windows and Macintosh are now commonly used. Though many useful GUI applications are existing now, there are few proposals to adopt isolated word recognition to existing applications. Some speech recognition systems treat the names of icons and menu descriptions as the voice commands, but they use speech only as the replacement of GUI operation and do not make good use of spoken dialog.

Some strategies are important to achieve the effective speech interface; such as "How should we organize and select the voice commands?" "How should we choose the speech output?" and so on. Though such problems are task-dependent and have usually been

solved individually. But it is possible to standardize the speech interface, if we have definite principles to use speech.

Usually, GUI systems are created with common GUI components such as push buttons, list-boxes, pull-down menus, etc. People can input values or confirm processes with dialog boxes, which consists of some GUI components. Dialog boxes may be compared to the conversations with computers. It gives a set of information to operate step-by-step, so this set is convenient for both GUI and spoken dialog.

World Wide Web also includes, a standard of GUI components and dialog boxes. Popular web browsers support "forms" and common gateway interfaces (CGI). So the browser show the GUI components and the user can access on-line databases with it.

Using CGI, web browser acts as a task-independent application client. If we want to add speech input/output feature to some stand-alone application, we have to make many modifications to the original software. On the other hand, we can add speech interface to the web applications easily. The client system needs some modifications, but the web server can be preserved. Such a system can help users who want to use speech with conventional GUI.

We propose a system that retrieves information from the Internet using speech input and output. This system uses only the standard protocols such as HTML and CGI. So, we can access all the database systems which support Hyper Text Transfer Protocol (HTTP) and CGI protocols.

The advantages of this approach are followings:

- The client system can access to existing on-line databases.
- The database server can provide both graphical user interface and speech enabled interface with the same protocols.
- The dialog control information is free from natural languages of text, because it is extracted from mark-up tags. If the system is combined with automatic language translation system, it enables cross-language queries.

There have been many attempts and products to apply speech input to web browsing. In many of such systems, words on the display are treated as voice commands if the words have links to other pages. Such a design is not so attractive as an application of speech interface, because the usual GUI operations with a mouse and a keyboard are very familiar and users sometimes feel they are more convenient than speech.

Our goal is not only to improve the operations of web browsers using speech interface, but also to realize a system which enables the users to do their task efficiently through natural spoken dialog.

## Design Policy of Spoken Dialog Systems

A spoken dialog system is considered as the ultimate solution to get rid of a barrier between human and machine. It is especially useful for information retrieval applications and on-line transactions.

There are many demonstration systems and some commercial systems achieving such features. Most of them, however, ignore the presence of graphical user interfaces, which greatly improved in these days. Most personal computer users are accustomed to desktop metaphors, WYSIWYG (What you see is what you get), pull-down menus and dialog boxes.

Spoken dialog systems, therefore, are not so attractive if there are no considerations to integrate speech input/output and current GUI environment.

Recently, speech recognition systems for isolated words have become very robust. The accuracy of voice command recognitions have become to satisfy the people's demand, but many researchers and application programmers do not create smart applications of speech input, so they prevent the utilization of speech input systems.

In particular, most speech applications use voice commands excessively, even if a mouse or a keyboard is more convenient. On the contrary, a spoken dialog system can save the labor of users if the utility of speech input is considered in multimodal input environment.

Many designers of interface have proposed some guide for interface design. One of the authors suggested basic principles of interface and organization principles of interface, classifying the basic requirements in interface systems and desired organizations that match various levels of users' skill (Nishimoto et al. 1995).

The principles consist of *Basic Principles of Interface* and *Organization Principles of Interface*.

### Basic principles of interface

Basic principles of interface indicate the required functions for a comfortable input system.

#### 1. Principles of Less Operations

- a) Minimize the motion of position;  
The movement of hands, fingers and a mouse pointer should be minimized.

- b) Minimize the number of operations;  
The number of operations required for a command should be minimized.
- c) Avoid input errors and unexpected acceptances;  
The system should avoid input errors and unexpected acceptances of command.

#### 2. Principles of Ease in Learning Operations

- a) Make easy to remind the command;  
Command names should be easy to remember. For this purpose, command names should be associated with their functions and not be confusing with other commands.
- b) Keep consistency;  
All commands should be operated with consistent methods within an application and across many applications.

#### 3. Principles of Transparency

- a) Make easy to perceive the status;  
The status of a system and available operations should be understood easily.
- b) Make easy to estimate results;  
The result produced by an operation should be always predictable.
- c) Offer perfect feedback;  
Perfect feedback should be always given for operations.

## Organization Principles of Interface

On which of these principles the emphasis should be placed depends on how long time and how frequently the user experienced the system. To realize an input system that is welcomed by users at various levels of skill, a system should provide the organization which satisfies both beginners and experienced users.

Organization principles of interface guide us that how each basic principle should be considered while we organize a system to meet the demand of each level of skill.

- a) Principle of supporting beginners;  
Every input system of application should have common styles for those who are unfamiliar with the system.
- b) Principle of treating frequent users with priority;  
Every input system should provide the specialized usages of an application which give the experienced users more efficiency than the operations for beginners.
- c) Principle of suggesting beginners to use skilled usage;  
Every input system should have guidance mechanisms that indicate skilled usage to inexperienced users.

From the viewpoint of the two sets of principles, speech input should improve the interface, especially on quantity of operation.

## Task Independent Design for Multimodal Dialog System

In making a multimodal dialog system, it is important to choose design that fits the domain of application and the task. For example, a multimodal drawing tool should be designed in such a way that the mode can be changed by speech input so that the user can use mouse only to draw figures (Nishimoto et al. 1995).

Before describing the strategy we adopt in the spoken dialog system proposed in this paper, we will outline how to treat speech in multimodal systems so far studied.

Usually most researchers of spoken dialog systems decide a first what kind of speech application is useful, then they build the systems. In many cases, the systems are well adopted for the task and useful (Smith and Hipp 1995).

The speech interface module of the system is, however, customized to the application, so it is often too difficult to modify it to work for another application.

General purpose multimodal dialog systems have been also proposed, which has authoring tool or user-interface description language (Matsu'ura et al. 1994). The application programmer who uses such systems can create a dialog system with GUI and reusable speech interface components. The advantage of this approach is that the utility of speech is fully optimized for some task, because the design of each dialog is described individually. Suppose, for example, there be a situation where general users easily remind some word as a voice command. If the word is not confused with the other command words, and the operation associated with the word is easily predicative, the programmer can add the word to the voice command list.

The disadvantage is the cost of starting again from the interface design of existing GUI applications. At the sacrifice of full optimization for certain application, it is possible to build a general purpose spoken dialog system, which extends existing GUI design and is optimized for every task to some degree. We previously described some interface principles. This approach does not apply the principles to the whole system, but applies them to the each of the minimum components which are used in the GUI application.

### Dialog Box As A Spoken Dialog Unit

The spoken dialog system proposed in this paper uses only the standard protocols such as HTML and CGI. Here we will give an example of how to control a dialog using such standard protocols.

GUI systems have a component called "dialog box." It is a small window, including check boxes, radio buttons, text input boxes, list boxes, with the notation which shows the meaning of values of each component. When the system displayed a dialog box, the user has only to finish the dialog and other operations are prohibited. The user has to pay attention to each message in the dialog box, input or modify some values,

and push OK or CANCEL button to confirm the input (Figure 1).

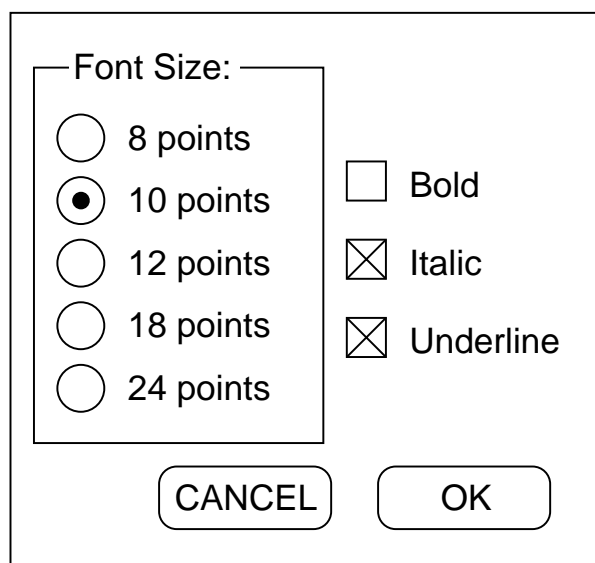


Figure 1: An example of dialog box. This dialog includes radio buttons, check buttons, and push buttons.

We treat one dialog box as one session of multimodal dialog, and apply the speech inputs and outputs to each dialog box. If there are important values, which the user must input or make decision, the system should press the user for an answer. If the dialog box contains some words which are related to the changeable values, the system adds the words to the voice command list (Figure 2).

```
SYSTEM: "Select font size"  
USER: "10 points"  
SYSTEM: "You can choose bold, italic or underline"  
USER: "Italic and bold"  
SYSTEM: "Are you OK to proceed?"  
USER: "Yes"
```

Figure 2: An example of imaginary spoken dialog, based on the dialog box. The system asks the user to fill each blanks and finally confirms to the user.

### Task Independent Dialog System Using CGI

This approach can be applied to every GUI application. But if an application is divided into the interface part and the application server, it is easy to build portable

dialog management system, which accesses the same application servers.

We use CGI protocol to realize it. Both the ways of database access and the user utterance expectation are obtained from the mark-up files (Figure 3). The system decides the words to recognize and the messages to say from the texts that the form includes, and the system accesses to the server using the parameters in the tags.

## Dialog Management Strategies

In this section, we describe the main strategy in our spoken dialog system.

When a dialog box is shown on the display, the dialog system requests the user to fill the adequate values to finish the current dialog. Using the system, the user can take the advantage of speech interface.

More than one dialogs (CGI forms) can exist in one web page. There are also some words that are not related to the CGI access, but to be linked to other pages. The system controls the dialogs between the system and its user according to the following strategies.

### Basic Strategy

**System Utterances** First, the system asks the user to select one target from the all CGI forms on the display. Then, it asks the user to fill all the blanks in the form. If all the blanks are filled, the query is performed.

- If there are more than one forms on the screen at the same time;  
The system has multiple dialog targets. In this case, the system asks the user to choose one form as a target.
- If the system has one dialog target (the user has already chosen one, or the page has one form);  
The system gives the order of priority to each item in the form, based on the boldness and displayed order of the item.  
The system utterances are proceeded in the order of priority, asking the user to fill the blank items. Speech output is used for the purpose of making easier to understand and decreasing the time to finish the task.
- If all the items in the form is already filled;  
The system makes a confirmation, something like “May I show you the results?” If the next utterance of the user is “Yes”, it will be recognized as the request of query.
- If the page contains only links and no dialog (form);  
The system asks the user to choose one item from the links.

**User Utterance Expectation** The followings are the utterances of the user that the system expects, and the actions the system should take in each case.

- A word which is linked to other pages;  
The system shows the destination of the link.
- A voice command which appeared in the former dialogs;  
The system goes back to the status where the word appeared, then the voice command is processed in the dialog.
- The names of items in the form;  
They are always accepted as the voice command, whether the item is filled or still blank. Once the item is filled by the user, the system does not ask the user to input it again.

### Strategy for Multiple Dialogs

Speech input can provide shortcut commands which are easy to remember.

Let's assume that a sequence of dialogs be required to finish a task. If the user wants to go backward to some steps and correct a value, the user would repeat “cancel” operation until he/she finds the dialog which is wanted to change. In such a case, if the speech commands are organized properly, the same operations can be done with one voice command. It saves the time and labor, and it is very easy for the users to learn the operations (Figure 4).

To realize this kind of operation, the system should remember the keywords which appeared in former dialogs, and always accept them as the voice commands.

### Priorities of the Topics

In a spoken dialog system, one dialog session has one or more topics. The system computes the priorities of each topic, because if the system takes the initiative in the dialog session, the system must know in what order the question should be done.

In usual GUI design, important questions are written boldly or strikingly. Such items are arranged on top of the dialog box, the size of letters is enlarged, or the messages are prominent. Our strategy decides the priorities of each word by analyzing such attributes of the text shown on the display; if the size of characters on the display is large, the item is regarded as an important item.

### Case Studies

We make some investigations on some details of proposed strategies.

- If the dialog box contains the item which can not display all choices at the same time (Figure 5), the user has to push the listbox and to hold the mouse button to see the options. In such case, if the system talks “What do you want, A, B, or C?” , the user will know what to input next, or what options he can choice. It meets the demand of *Principles of Less Operations* and *Principles of Transparency* previously mentioned.

```

<form>
<input type="radio" name="fontsize" value="8" > 8 points
<input type="radio" name="fontsize" value="10"> 10 points
<input type="radio" name="fontsize" value="12"> 12 points
<input type="radio" name="fontsize" value="18"> 18 points
<input type="radio" name="fontsize" value="24"> 24 points

<input type="checkbox" name="bold" value="1"> Bold
<input type="checkbox" name="italic" value="2"> Italic
<input type="checkbox" name="underline" value="3"> Underline
<input type="submit" value="OK">
</form>

```

Figure 3: An example of the form written in HTML. This dialog includes 5 radio buttons and 3 check buttons. In this case, the “name” and “value” parameters are used to access the server.

- In the case there are many choices, if the system reads all the name of options, it takes long time and users get tired of it (Figure 6). All the user has to know is that the input of this item cannot be omitted. The options themselves are easy to see on the display. In this case, “Please read the display and choose one option” will be more appropriate as the system message.
- If there are many options on the display, and they are not so essential, the system does not demand the input. The system only accepts the items as the voice input command. When the user wants to choice some words, he can only to say the word, or select by the mouse (Figure 7).

### Cross-Language Extensions

Dialog management policy of the system depends on the form tags in the original GUI system, and does not depend on the messages itself. Therefore, the system is free from what a language the text is written in.

There are many commercial machine translation systems aimed at web browsing. We can use such a translation system as the pre-process of the dialog management. So, if the target database system inputs or outputs messages in other language than the native one of the dialog system, the system needs no modifications in dialog management, on the assumption that the machine translation system generates accurate results.

### Configuration of Proposed System

Figure 8 shows the organization of proposed system.

#### Machine Translation

As a language translation system, we use existing commercial software. At present we use English-Japanese MT system, because our system only performs Japanese dialog management.

#### Dialog Target Analysis

It analyzes the HTML tags and computes the priority of each topic as previously mentioned.

#### Dialog Management

It selects the item with the highest priority and asks the user to fill the blanks using speech output. At the same time, it also generates expectations of user utterances.

#### User Event Management

It controls the speech recognition system, the speech synthesis system, and the web browser. It manages user events and gives multimodal feedbacks to the user.

#### Web Browser

User event manager sends HTML data to the web browser. To give the real-time feedbacks to the user, GUI components in original HTML files are replaced by Java applets, so they can communicate with the event manager.

#### Speech Synthesis/Recognition

Speech synthesizer is a commercial text-to-speech synthesizer. The speech recognition module uses phoneme-based Hidden Markov Models (HMM). Japanese words in HTML files are converted to the sequence of phonemes and used for the expectation of the user’s utterances.

### Conclusion

We proposed a practical design of a general purpose spoken dialog system, and applied to the on-line database services on the WWW. This enables task-oriented spoken dialogs with a single task-independent client software.

As the future work, we are interested in the evaluations of the system. The dialog management module may need some adaptations to the each user’s skill, or the reliabilities of the speech recognition.

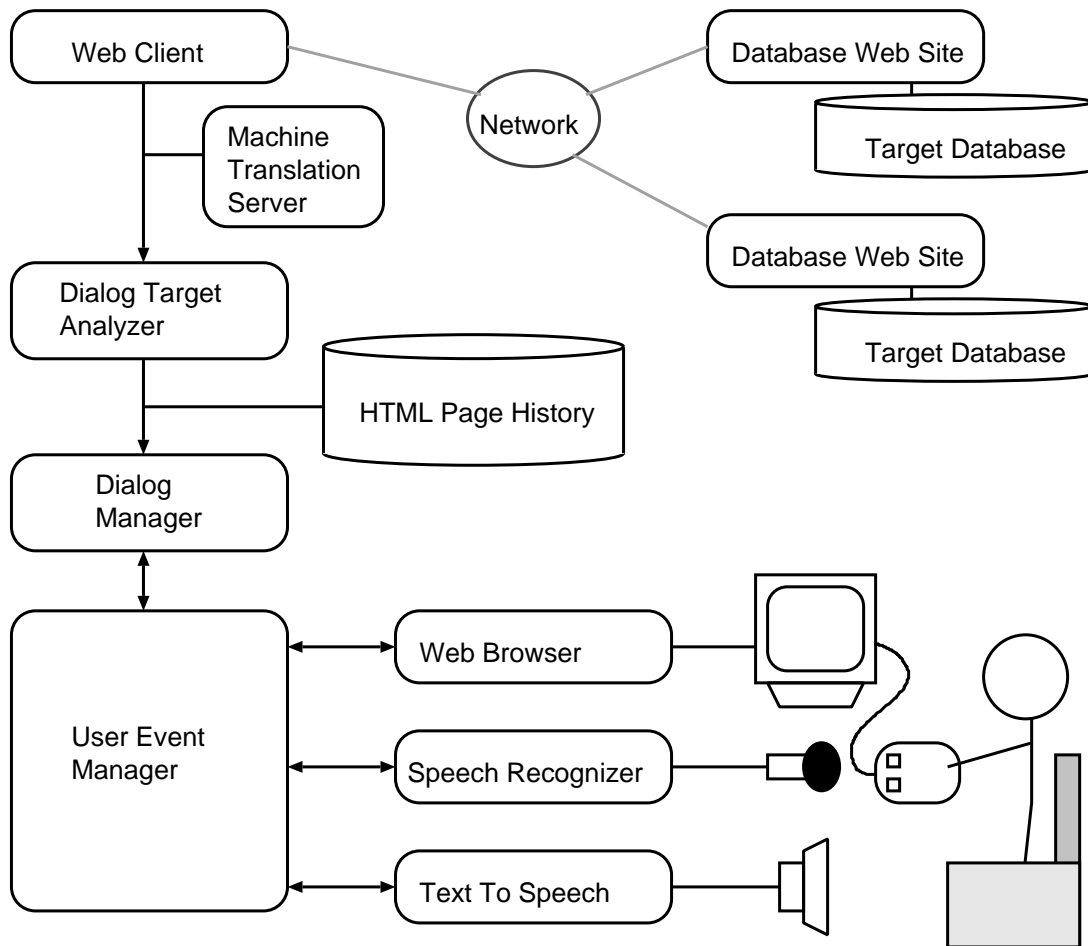


Figure 8: Organization of the system.

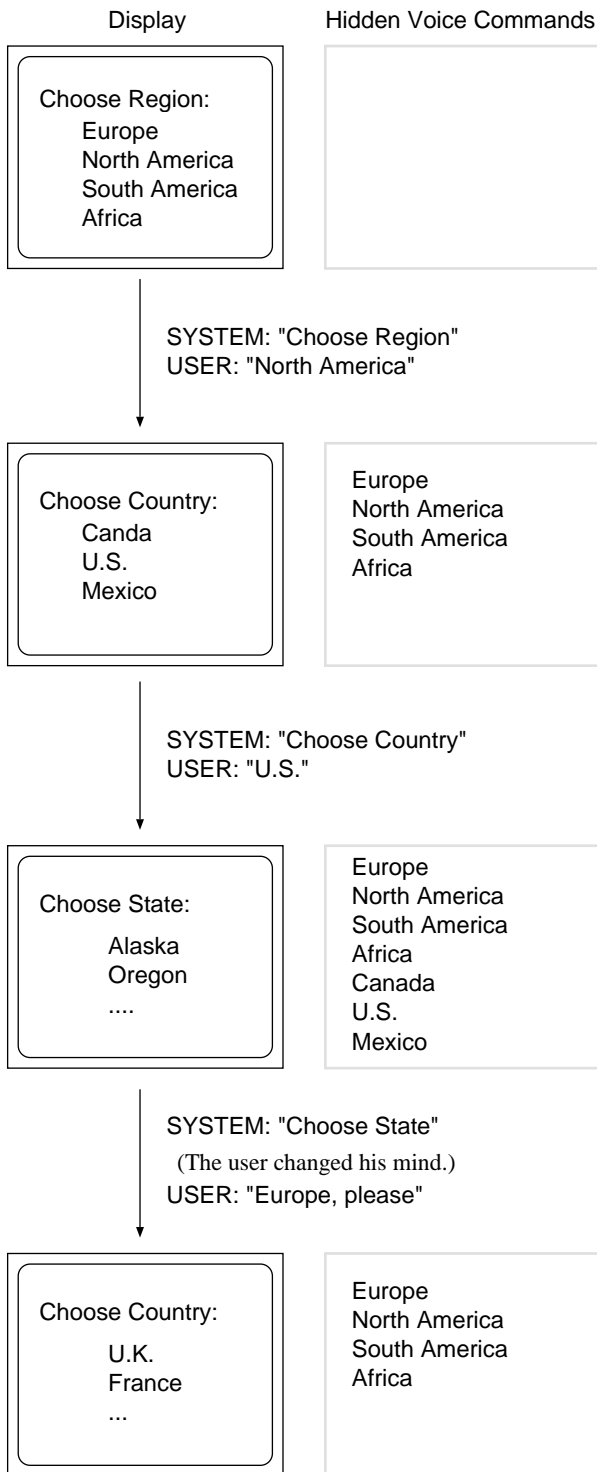


Figure 4: A sample case of applying dialog strategy among pages. The user can speak the words in the display (left boxes in this figure) and also hidden voice commands (right boxes in this figure) in the former dialogs.

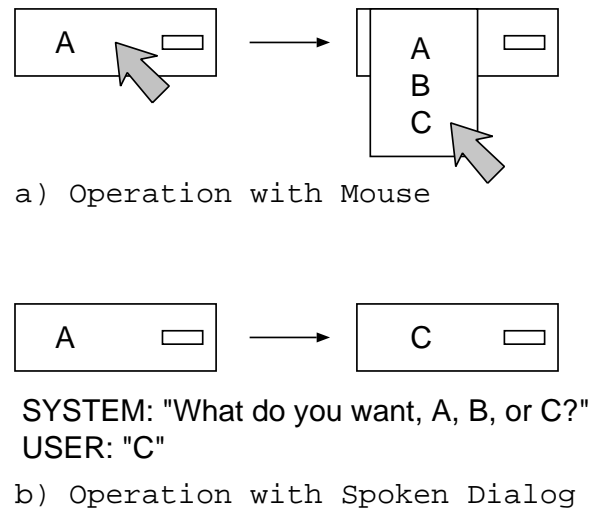


Figure 5: A sample case of applying dialog strategy to a dropdown box. a) Without speech, mouse operations are required to know the options. b) With speech, the system helps the user to perceive the status.

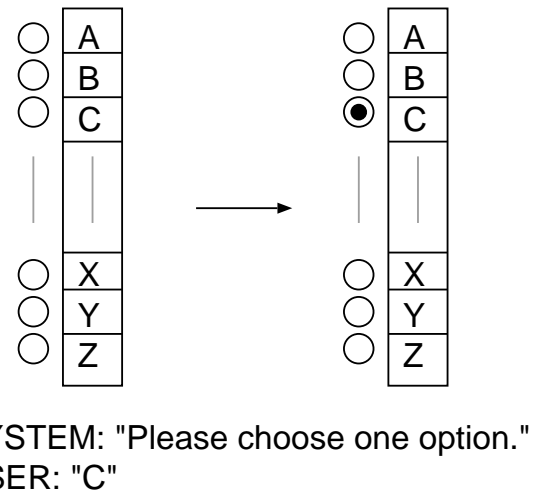


Figure 6: A sample case of applying dialog strategy to radio buttons. In this case, there are many options, so the system does not read every item.

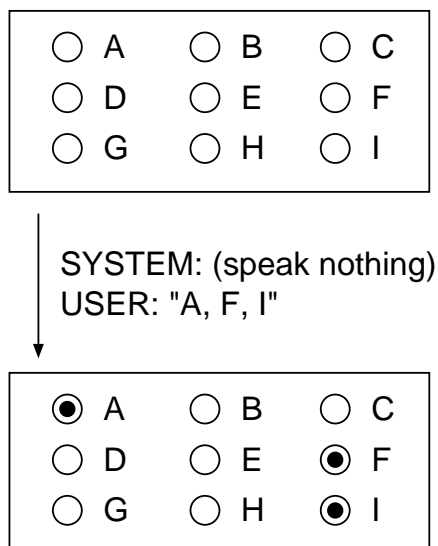


Figure 7: A sample case of applying dialog strategy to check boxes. These items are optional, so the system speaks nothing about them.

### References

- Nishimoto, T.; Shida, N.; Kobayashi, T.; Shirai, K. 1995. Improving Human Interface in Drawing Tool Using Speech, Mouse and Key-Board. *Proceedings of 4th IEEE International Workshop on Robot and Human Communication, ROMAN95* : 107-112. Tokyo, Japan.
- Smith, R.W. and Hipp, D.R. 1994. *Spoken Natural Language Dialog Systems: A Practical Approach*. Readings, Oxford University Press.
- Matsu'ura, H.; Masai, Y.; Iwasaki, J.; Tanaka, S.; Kamio, H.; Nitta, T.; A Multimodal, Keyword-based Spoken Dialogue System – MultiksDial. *Proceedings of ICASSP'94 II* : 33-36. Adelaide, South Australia.