Drawing Pictures with Natural Language and Direct Manipulation

Mayumi Hiyoshi and Hideo Shimazu
Information Technology Research Laboratories, NEC Corporation
4-1-1 Miyazaki, Miyamae Kawasaki, 216 Japan
{hiyoshi, shimazu}@joke.cl.nec.co.jp

Abstract

A multimodal user interface allows users to communicate with computers using multiple modalities, such as a mouse, a keyboard or voice, in various combined ways. This paper discusses a multimodal drawing tool, whereby the user can use a mouse, a keyboard and voice effectively. Also, it describes an interpretation method, by which the system integrates voice inputs and pointing inputs using context.

1 Introduction

This paper describes an experimental implementation of a multimodal interface. Specifically, the authors have developed a multimodal drawing tool. The multimodal drawing tool allows users to draw pictures by using multiple modalities; mouse, keyboard and voice input, in various combined ways.

Recently, most user interfaces tend to be based on a direct manipulation method. However the direct manipulation method is not always better than other ways. The direct manipulation method is not particularly applicable, when mentioning several operations together and operating an object which is not displayed. Also, it compels a user to point to a target object correctly with a pointing device. On the other hand, voice inputs have some advantages, since a user can feel free to speak at any time, and the user can use the voice input while simultaneously using other devices. A combination of such different modalities offers an interface which is easy for the user to use.

Many multimodal systems, which integrate natural language inputs and pointing inputs, have been developed [2][1][5][4]. In those systems, the user uses natural language mainly supported by the pointing inputs. However, when the user has to communicate with the computer frequently, in such a system as drawing tool, it is not effective for the user to always speak while working.

A prototype system for a multimodal drawing tool has been developed, whereby the user can use voice inputs unrestrainedly and effectively, that is, the user can choose a modality unrestrainedly, and can use the voice inputs only when the user wants to do so. In such a system, input data come in at random with multiple modalities. The multimodal system must be able to handle such several kinds of input data.

2 Multimodal Inputs in Drawing Tools

This section describes requirements to develop general drawing interfaces. In existing drawing tools, a mouse is a major input device. In addition, some drawing tools assign functions to some keys on a keyboard to reduce inconvenience in menu operations. Issues regarding such interfaces are as follows:

- It is troublesome to input a function. Because a user uses a mouse, both to select a menu and to draw a figure, the user has to move a cursor many times from a menu area to a canvas on which figures are placed.
- It is troublesome to look for a menu item. In proportion to increasing functions increment, menu items also increase. So, it becomes increasingly difficult to look for a specific objective menu item.
- It is troublesome to continuously move a hand from a mouse to a keyboard.
- It is not possible to express plural requirements simultaneously. For example, when a user wants to delete plural figure objects, the user has to choose the objects one by one.
- The user has to point to an object correctly. For example, when the user wants to choose a line object on a display, the user has to move a cursor just above the line and click the mouse button. If the point shifts slightly, the object is not selected.

By adding voice input functions to such an input environment, it becomes possible to solve these first

three issues. That is, by means of operation with the voice input, a user can concentrate on drawing, and menu search and any labor required by changing devices becomes unnecessary.

For overcoming the rest of these issues, more contrivance is needed. The authors attempted to develop a multimodal drawing tool, operable with both voice inputs and pointing inputs, which has the following functions.

- A user can choose a modality (mouse or voice) unrestrainedly, which means that the user can use the voice inputs only when the user wants to do so. Also, the user can use both modalities in various combined ways. For example, the user says "this", while pointing to one of several objects.
- Plural requests can be expressed simultaneously (ex. "change the color of all line objects to green"). So, the operation efficiency will be improved.
- A user can shorten voice inputs (ex. "move here") or omit mouse pointing events based on the situation, if the omitted concepts are able to be inferred from context. For example, the user can utter "this", as a reference for previously operated objects, without a mouse pointing.
- Ambiguous pointings are possible. When a user wants to choose an object from among those on a display, the user can indicate it roughly with a brief description, using the voice input. For example, a user points at a spot near a target object and utters "line", whereby the nearest "line" object to the spot is selected. Or, a user points at objects piled up and says "circle", then only the "circle" objects among the piled up objects are selected.

To realize these functions, it is necessary to solve the following new problems.

1. Matching pointing inputs with voice inputs.
   In the proposed system, since pointing events may often occur independently, it is difficult to judge whether or not an event is an independent input or whether it follows a related voice input. So, an interpretation wherein the voice input and pointing event are connected in the order of arrival is not sufficient [4]. Therefore, a pointing event should be basically handled as an independent event. Then, the event is picked out from input history afterward, when the system judges that the event relates to the following voice input.
2. Solving several input data ambiguities.
   In the previous mouse based system, ambiguous inputs do not occur, because the system requires that a user selects menus and target objects explicitly and exactly. Even if the voice input function is added in such a system, it is possible to force the user to give a detailed verbal sequence for the operation without ambiguity. However, when the function becomes more sophisticated, it is difficult for the user to outline the user's intention in detail verbally. So, it is necessary to be able to interpret the ambiguous user's input.

Several multimodal systems have been developed to solve these problems. For example, Hayes [4] proposed the first issue to be addressed, but the definite solution was not addressed. Cohen [3] presented a solution for the first issue, by utilizing context. However, the solution is not sufficient for application to drawing tools, because it was presented only for query systems. The following section describes a prototype system for a multimodal drawing tool. Next, solutions for these problems are presented.

3 Multimodal Drawing Tool

3.1 System Construction

A prototype system for a multimodal drawing tool was developed as a case study on a multimodal communication system. Figure 1 shows a system image of the prototype system, by which the user draws pictures using mouse, keyboard and voice. This system was developed on a SUN workstation using the X-window system and was written in Prolog. Voice input data is recognized on a personal computer, and the recognition result is sent to the workstation.

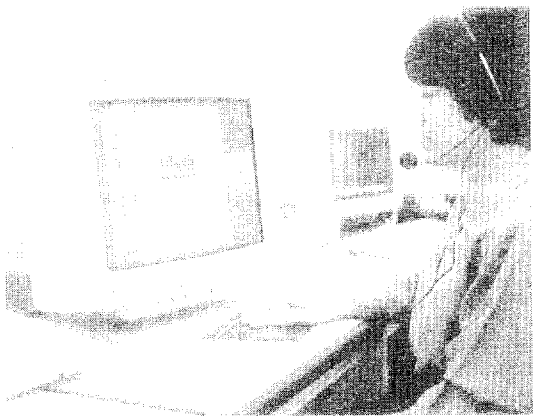

Figure 1: System Image

3.2 Interface Examples

Figure 2 shows a screen image of the system. The user can draw pictures with a combination of mouse and voice, as using a mouse only. Input examples are follows:

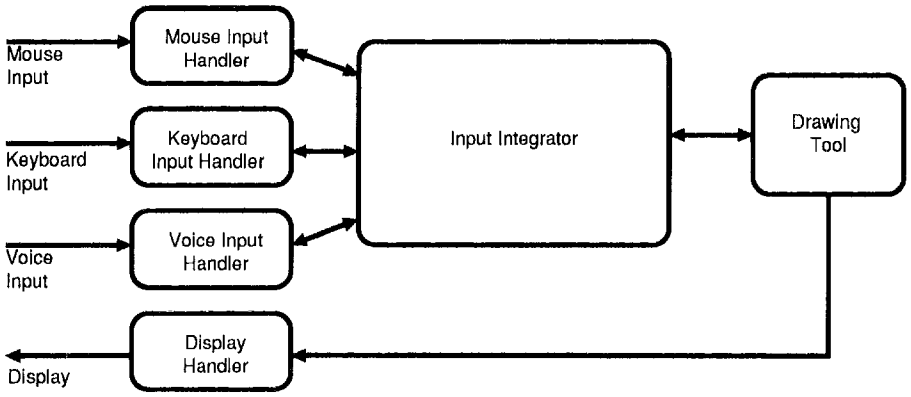

Figure 3: Multimodal Drawing Tool Structure

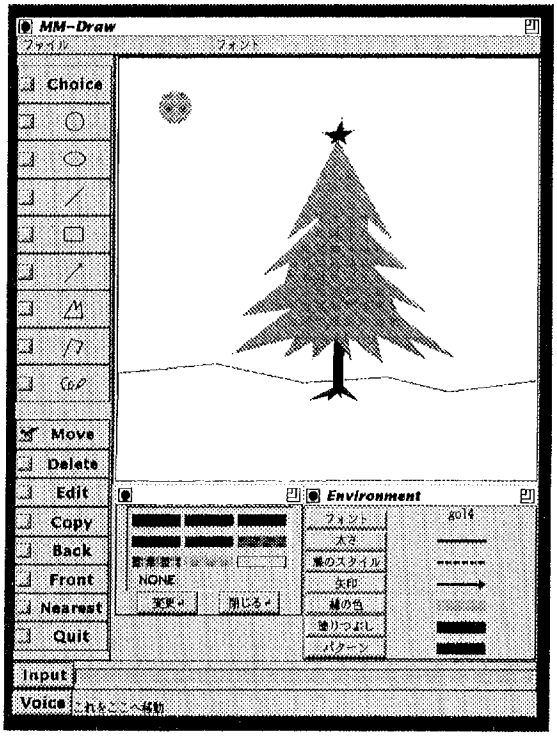

Figure 2: Screen Image

- If a user wants to move an existing circle object to some point, the user says "Move this circle here", while pointing at a circle object and a destination point. The system moves the circle object to the specified point.
- If a user wants to choose an existing line object among several objects one upon another, the user can say "line" while pointing at a point near the line object. The system chooses the nearest line object to the point.
- If the user wants to draw a red circle, the user can say "red circle". The system changes a current color mode to red and changes the drawing mode to the circle mode.

3.3 System Structure

Figure 3 shows a prototype system structure. The system includes Mouse Input Handler, Keyboard Input Handler, Voice Input Handler, Drawing Tool and Input Integrator.

Each Input Handler receives mouse input events, keyboard input events and voice input events, and sends them to the Input Integrator.

The Input Integrator receives an input message from each input handler, then interprets the message, using voice scripts, which show voice input patterns and sequences of operations related to the patterns, as well as mouse scripts, which show mouse input patterns and sequences of operations. When the input data matches one of the input patterns in the scripts, the Input Integrator executes the sequence of operations related to the pattern. That is, the Input Integrator sends some messages to Drawing Tool to carry out the sequences of operations. If the input data matches a part of one of the input patterns in the scripts, the Input Integrator waits for a next input. Then, a combination of previous input data

and the new input data is examined. Otherwise, the interpretation fails. The Input Integrator may refer to the Drawing Tool. For example, it refers to the Drawing Tool for information regarding an object at a specific position.

The Drawing Tool manages attributes for figure objects and current status, such as color, size, line width, etc. Also, it executes drawing and editing operations, according to requests from the Input Integrator, and it sends the editing results to the Display Handler. The Display Handler modifies the expression on the display.

4 Multimode Data Interpretation

This section describes in detail interpretation methods for the multimodal inputs used in the drawing tool.

4.1 Matching Pointing Inputs with Voice Inputs

In conventional multimodal systems, all anaphoric references in voice inputs bring about pointing inputs, and individual pointing input is connected to any anaphoric reference. However, in our system, a user can operate with either a pointing input only, a voice input only or a combination of pointing events and voice inputs. Because a pointing event may often occur independently, when a pointing event does occur, the system cannot judge whether the event is an independent input or whether it follows the related voice input. Furthermore, the user can utter "this", as reference to an object operated immediately before the utterance. So, an interpretation that the voice input and pointing event are connected only in the order of arrival is not sufficient. In the proposed system, a pointing event is basically handled as an independent event. Then, the event is picked out from input history afterward, when the system judges that the event relates to the following voice input. Furthermore, the system has to interpret the voice inputs using context (ex. previous operated object).

In the proposed system, pointing inputs from start to end of a voice input are kept in a queue. When the voice input ends, the system binds phrases in the voice input and the pointing inputs in the queue. First, the system compares the number of anaphoric references in the voice input and the number of pointing inputs in the queue. Figure 4 shows timing data for a voice input and pointing inputs. In Case(1), the number of anaphoric references in the voice input and the number of pointing inputs. In the other cases, a pointing input is lacking. When a pointing input is lacking, the following three possible causes are considered.

- The relative pointing event occurred before the voice input, and it was handled previously (Case(2) in Fig. 4).
- The first anaphoric reference is "this" as reference to an object which was operated immediately before the voice input (Case(3) in Fig. 4).
- The relative pointing event will occur after the voice input (Case(4) in Fig. 4).

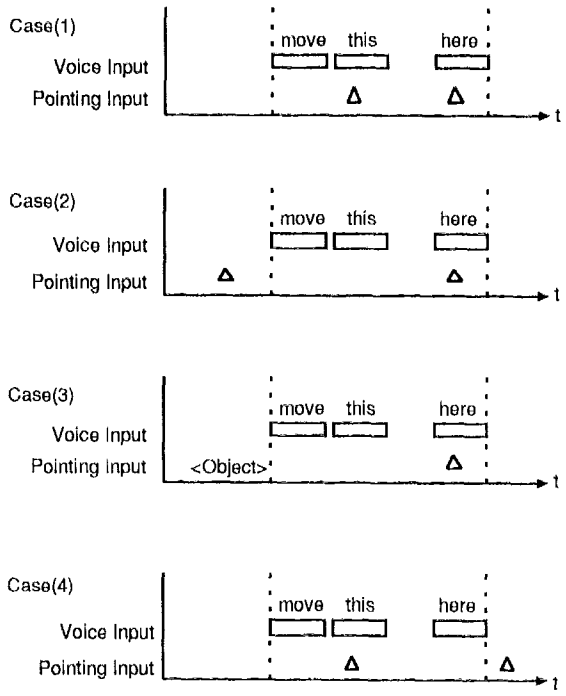

Figure 4: Timing data for voice input and pointing inputs

The interpretation steps are as follows.

1. The system examines an input immediately before the voice input. If it is a pointing event, the event is used for interpretation. That is, the event is added at the top of the pointing queue.
2. When the above operation fails and the first anaphoric references is "this", then the system picks up the object operated immediately before, if such exists. The object information is added at the top of the pointing queue.
3. Otherwise, the system waits for the next pointing input. The input is added onto the last of the pointing queue. When a time out occurs, the interpretation fails, due to a lack of a pointing event.

If the system can obtain the necessary information, it binds the anaphoric references in the voice input and pointing event and object information in the pointing queue in the order of arrival.

4.2 Solving Input Data Ambiguity

In a conventional mouse based system, there is no semantic ambiguity. Such systems require a user to select menus and target objects and to edit the objects explicitly and exactly. Even if the voice input function is added in such a system, the user can be forced to utter operations without ambiguity. However, when the function becomes more sophisticated, it is difficult for the user to utter the user's intentions in detail. So, it is necessary to be able to interpret the user's ambiguous input. In a multimodal drawing tool, such our system, one of the most essential input ambiguities is led by ambiguous pointings.

For example, if a user says "character string", there are three possible interpretations: "the user wants to edit one of the existing character strings", "the user wants to choose one of the existing character strings" and "the user wants to write a new character string"

In this example, the system interprets using the following heuristic rules.

- If a pointing event does not exist immediately before, the system changes a drawing mode to the character string input mode.
- If a pointing event upon a string object exists just before the voice input, then the system adds the character string object to a current selection; a group of objects selected currently.
- When a pointing event exists immediately before the voice input and there is a character string object near the position of the user's point (ex. within a radius of five mm. from the position), then the character string object is added to a current selection.
- When a pointing event exists and there is no character string object near the position, then the mode is changed to the character string input mode at the position.

Naturally, "character string" in these heuristics rules can be replaced by other figure types. If this heuristic rule is not perfect, the interpretation may be different from the user's intention. In such a case, it is important for a user to return from the error condition with minimum effort. For example, assume that a user, who wants to choose one of "character string" objects, says "character string" and points on the display, but the distance between the pointed position and the "character string" object is greater than the predefined threshold. Then, according to the above rules, the result of the system's interpretation will be to input a new character string at the position, and the drawing mode changes to the character string input mode. In this case, the user wishes to turn back to a state which the user intended with minimum effort. The system must return to the state in which the character string input mode is canceled and the nearest "character string" object is selected. A solution is for the user to utter "select" only. Then, the system understands that it's interpretation was wrong and interprets that "select" means "select a character string object" using current context.

5 Conclusion

For implementing a multimodal system, based on direct manipulation system, the system has to use not only pointing events concurrently with a voice input, but must also use the context, such as input history or information regarding the current operated object, as information for binding to the voice input. Furthermore, it is important to solve any ambiguity in inputs. This paper discussed these problems, and described an interpretation method using a drawing tool example. Furthermore, a prototype system for a multimodal drawing tool has been implemented. Much future work remains, but we believe that these elaborate interpretations may become bases of user friendly multimodal interfaces.

Acknowledgements

A part of this study was conducted under the FRIEND21 national study project.

References

[1] Allgayer, J., Jansen-Winkeln, R., reddig, C., and Reithing N., "Bidirectional use of knowledge in the multi-modal NL access system XTRA", IJCAI'89, pp.1491-1497, 1989.

[2] Bolt, R.A., "Put-That-There: Voice and Gesture at the Graphics Interface", Computer Graphics 14, 3, 1980.

[3] Cohen, P.R., Dalrymple, M., Moran, D.B., Pereira, F.C.N., et al., "Synergistic Use of Direct Manipulation and Natural Language", Proc. of CHI-88, 1989.

[4] Hayes, P.J., "Steps towards Integrating natural Language and Graphical Interaction for Knowledge-based Systems", Advances in Artificial Intelligence - II, Elsevier Science Publishers, 1987.

[5] Wahlster, W., "User and discourse models for multimodal communication", in J.W. Sullivan and S.W. Tyler, editors, Intelligent User Interfaces, chapter3, ACM Press Frontiers Series, Addison Wesley Publishing, 1989.