"Kairai" - Software Robots Understanding Natural Language

Yusuke Shinyama Takenobu Tokunaga Hozumi Tanaka

Department of Computer Science, Tokyo Institute of Technology
{euske, take, tanaka} @cl.cs.titech.ac.jp

Abstract

In this paper we give an overview of the "Kairai" virtual actor system, which understands natural language instructions and displays the results via software robots. In controlling software robots acting in a 3-D world, the system needs to access various types of information from language expressions. Here, we concentrate especially a handling anaphora, used to indicate objects or positions in the virtual world. Our system contains a robot belief database and analyzes the user's speech act in manipulating the database. We also consider ellipsis, which is used frequently in command-style dialogues.

1 Introduction

We are developing a virtual actor system named "Kairai", which has a Japanese speech interface allowing the use to manipulate software robots within a virtual world (Fig. 1). The user commands the software robots orally, and the software robots respond to the command in real time. The software robots can move forward, turn, or push an object. Manipulating the robots by commands, the user can move and place objects in the virtual world. Changes in the virtual world are presented to the user via animation.

Manipulating robots through natural language can be convenient in some situations. We believe that our system can be applied in various applications such as animation generation systems, graphic editors, and entertainment media. Controlling the camera by voice command in a virtual walk-through system or 3-D model viewer would facilitate convenient manipulation of the user's view. Furthermore, our technique could be applied to control real world robots with natural language.

Our system accepts the following input sentences, for example:

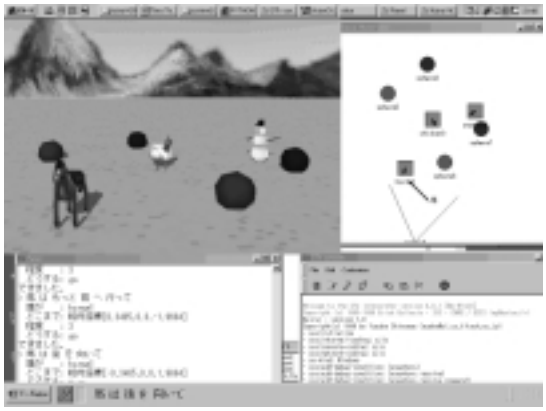

Figure 1: System screenshot

(1) "Niwatori wa sono kyu wo hidari kara osite."
(Chicken, push the sphere from the left.)
(2) "Mousukosi osite."
(Push it further.)
(3) "Yukidaruma mo sore wo osite."
(Snowman, push it, too.)
(4) "Camera wa sono mae ni aru kyu wo utsusite."
(Camera, show the sphere located in

front of it.)

(5) "Motto migi e."
(Further, to the right.)

There have been several attempts to realize this goal. The most famous one is Winograd's SHRDLU, which can understand various sentences and manipulate blocks in the block world [8]. In this kind of system, important research issues include identifying the action which the user intended, and the objects which the user specified. Our system can act as a testbed for experiments to verify theories on language and action in this paper. We focus on identifying referred objects using contextual information. In particular, we pay attention to ellipsis and anaphora frequently used in actual dialogue.

As seen in the examples above, the system can resolve ellipsis and anaphoric expressions, both of which are commonly observed in Japanese dialogue. The performance of ellipsis and anaphora resolution has a significant impact on the user friendliness of the system. In Japanese, ellipsis can be handled in the same manner as anaphora. Another significant feature of this system is the resolution of deictic expressions. Objects can be referred to with pronouns and definite noun phrases even when they do not have textual antecedents. The system can understand such exophora from the view of the software robots and the user.

In the following, first we present an overview of the system. Next we describe the anaphora resolution algorithm. To resolve anaphoric expression, the system needs to use various clues in user's utterance as well as knowledge of the virtual world. Finally, we describe future plans to extend this system.

2 System Overview

Our system consists of six modules (Fig. 2). The speech recognition module is provided with user voice input and converts

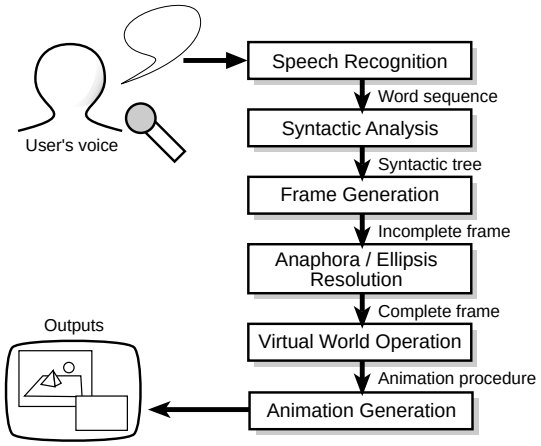

Figure 2: System Components

it into a word sequence. The word sequence is then syntactically analyzed and converted to a case frame structure representing the meaning of the utterance.

However, at this stage anaphora and ellipsis have not yet been resolved, that is, some slots may be left empty. Referents of pronouns and definite noun phrases are not identified at this stage either. This type of frame is thus called an "incomplete frame." The system resolves such deficiencies in latter stages in order to execute the user's instruction.

The incomplete frame is sent to the anaphora and ellipsis resolution module, which performs speech act analysis in order to determine the user's intention, and generate and modify robot belief models. Ellipsis and anaphora are resolved with reference to these belief models. The anaphora and ellipsis resolution module also uses a virtual world database to enable robots to understand deictic expressions.

After the anaphora and ellipsis resolution module, case frames are completed and sent to the virtual world operation module. This completed structure is called a "complete frame", in which each language expression is mapped to an object in the virtual world. The virtual world operation module retrieves and modifies the virtual world database, and generates a sequence of procedures that should be performed by robots. This data is sent to the

animation generation module in order to visualize changes in the virtual world via animation.

Each module generates all possible candidates at each stage for filtering in the next module. Although this methodology is computationally expensive, it makes the implementation simple. Here, the performance of the system depends on how many candidates each module can generate and how successful the next module is in selecting acceptable ones from among them. We can vary the system performance by modifying these constraints.

For example, the following sentence is converted into a case frame as in Fig. 3:

(6) "Niwatori wa kimi no mae ni aru kyu wo hidari ni osite."
(Chicken, push the sphere in front of you to the left.)

Since this sentence contains no ellipsis, the frame has all slots filled in order to execute the instruction. The system then identifies an object and its coordinates in the virtual world from this structure.

Our approach for semantic representation is similar to that of SHRDLU. Adjectives and modifying phrases are first converted into a predicate function to retrieve objects in the world. For example, the noun phrase "the sphere in front of you" is converted into the procedure:

$\lambda x.\lambda y.$ (y is a sphere $\wedge$ y is located in front of $x \wedge x$ is you).

This method works well when dealing with a finite number of objects. When identifying a location represented by a certain language expression, there would be infinite candidates because the coordinates of the virtual world constitute a continuum. We therefore use a "position generator" procedure to generate coordinates represented by the language expression. This procedure is also built from frame structures [7]. Sentence (6) is finally converted into a procedure operating on the world, as in:

```
Agent:
    Entity: Chicken
Object:
    Entity: the sphere
        Position: in front of
            Entity: you
To:
    Position: the left
Action: push
```

Figure 3: "Chicken, push the sphere in front of you to the left."

```
Agent:  (empty)
Object:
    Entity: the red sphere
Action: push
```

Figure 4: "Push the red sphere further."

```
push(chicken1, sphere4, [3, 0, 1]).
```

The sentence (7) following sentence (6) includes ellipsis (i.e. it has no subject) and a definite noun phrase ("the red sphere"):

(7) "Sono akai tama wo motto osite."
(Push the red sphere further.)

This sentence is converted into the frame given in Fig. 4. In this case, the system needs to resolve the ellipsis (agent slot) as well as identifying the object referred to as "the red sphere." In the next section, we describe the anaphora/ellipsis resolution algorithm, which fills up incomplete frames into complete ones.

The core parts of the system were written in the functional language Scheme [3]. The system runs under Microsoft Windows and uses IBM ViaVoice and Alice, the interactive graphics system [2]. These components communicate with each other by way of socket interfaces.

3 Resolving Anaphora

In resolving anaphoric expression, the notion of "focus" is important [4]. Focus suggests antecedent candidates of anaphors in a given sentence. Grosz et al. pointed out that task-oriented dialogues can be divided

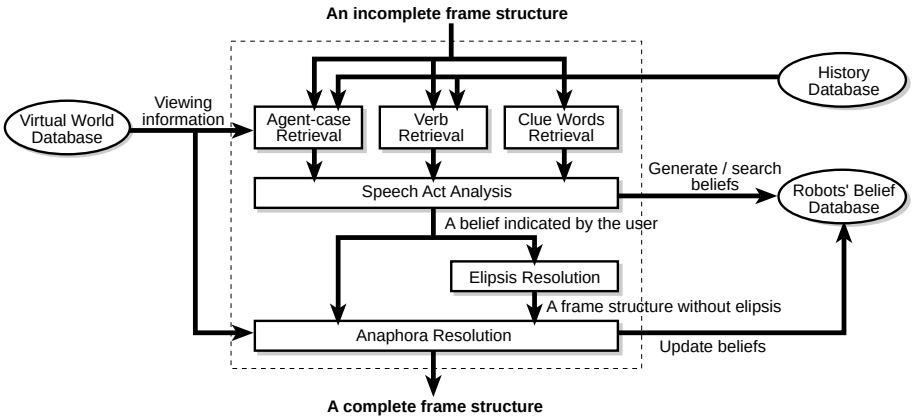

Figure 5: Anaphora and ellipsis resolution module

into subparts based on their purpose, and that the segment purpose affects the focus within that part [5]. In our system, the user delivers an action plan to be performed. So we suppose that the focus of dialogue varies as the action commanded by the user changes. Moreover, each robot has its own belief and acts based thereupon. We use this assumption in resolving anaphoric expressions.

Here we introduce the concept of "speech acts", the act intended by saying something [1]. Speech acts are not always marked explicitly as lexical and syntactic features but implied by various properties in the sentence. The user uses speech acts effectively to deliver his/her plan to software robots. Our system assumes two types of the user speech acts, "delivering a new instruction" and "modifying the belief (of software robot)." For example, sentence (6):

(6) "Niwatori wa kimi no mae ni aru kyu wo hidari ni osite."
(Chicken, push the sphere in front of you to the left.)

represents the "delivering a new instruction" speech act, whereas the sentence (7):

(7) "Sono akai tama wo motto osite."
(Push the red sphere further.)

represents the "modifying the belief" speech act. The system determines the type of speech act based on the surface linguistic forms. For example, when ellipsis or anaphora occur in a sentence, the "modifying speech act" is assumed. This type of speech act can also be suggested by clue words like "further", "too" and so on. In the above examples, the frame generated from sentence (6) has all its mandatory slots filled. On the other hand, sentence (7) does not. Additionally sentence (7) contains the clue word "further", which implies that the user is trying to modify the previous robot belief.

Our anaphora and ellipsis resolution module maintains a database containing the current belief model of each robot. Belief is represented by a frame structure indicating what to do. This module converts an incomplete frame to a complete one (Fig. 5). At the same time this module determines the user's speech act, identifies the robot's belief, and updates the database by replacing old beliefs with new ones. When the user provides a new goal, the module creates a new belief frame in the database. When the user provides a previously mentioned goal and if he/she wants to modify it, the module updates the database. The database also maintains the history of a dialogue, which is referred to in identifying speech act types.

When the module receives an incomplete frame, it searches the database to find a belief consistent with the frame. Consistency

is checked by comparing each slot in the frame. If no frame in the database matches the current frame, the system assumes that the user has provided a new goal. In this case, any instances of anaphora are considered simply to refer to the previous noun phrase. When a consistent frame is found, the system uses it as the robot's belief and copies it to the current frame. Only when an appropriate candidate for the anaphor is not found in the database, does the system consider it as a deictic expression and refer to the virtual world database in order to access the user's and robots' view. The system uses heuristic rules for this purpose. Finally, the system adds the modified frame into the database for latter processing.

We suppose that the agent and verb slots play an important role in specifying the action. When these two slots in the current sentence match with those of a frame in the history database, the system considers the current sentence to refer to the action specified in the matched sentence. It is sometimes possible to omit these slots without any linguistic clue, as to what is going on, as occurs with sentence (7). Here, the system must retrieve the missing slots from past sentences, or fill them with default values. In Fig. 5, the "Agent-case Retrieval" and the "Verb Retrieval" components carry out this task.

Consider the example of sentence (7) being uttered following sentence (6). In this case, sentence (7) does not include an agent slot. The Agent-case Retrieval component fills this slot with the previous value "Chicken" as the most possible candidate. The "Speech Act Analysis" component of this module can determine that the user's intention is "modifying the belief" and find the belief generated in sentence (6) which should be modified. This result is sent to the "Anaphora Resolution" component to resolve the anaphor "the red sphere". The Anaphora Resolution component uses the sphere referred to in sentence (6) as the object which should be pushed.

Furthermore, the user may utter even sentences such as:

(8) "Oshite." (Push.)
(9) "Motto." (Further.)

In these sentences, objects or verbs are omitted in addition to subjects. Kameyama pointed out that in Japanese these omitted noun phrases can be handled as "zero-anaphors", an invisible pronoun referring to an antecedent [6]. In our system, ellipsis is represented as a missing slot in a frame. The system fills up missing slots by copying slots from consistent robot beliefs found by the Speech Act Analysis component. Consequently, the above sentences are interpreted respectively as:

(8′) "[Sore wo] Oshite." (Push [it].)
(9′) "[Sore wo] Motto [Oshite]." ([Push] [it] further.)

When the system cannot find a referent in the preceding textual context, the Anaphora Resolution component regards the expression as deictic. In this case, the Anaphora Resolution component communicates with the virtual world database to get an information on objects which is within the robots and camera view. Thus, the system can handle cases where the user uses a pronoun to refer to an object located in the camera view, or in the front of the current robot.

4 Conclusion

In this paper, we described the anaphora and ellipsis resolution algorithm used in a virtual actor system. First the system converts the input sentence into a case frame. The system determines the user's speech acts from analysis of the robot belief database. Then the system identifies the indicated belief and uses it to complete the case frame, through the resolution of ellipsis or anaphoric expressions. The system can also resolve deictic expressions by referring to the virtual world database.

In future work, we plan to achieve the following goals:

- Handling of hierarchical structured objects, for example, a house with three rooms, each of which has two tables. This would allow the user to say something like "get the cup on table A in room 2".

- Handling of hierarchical goals and planning. Currently the system assumes that intentions are monolithic. But a dialogue may consist of hierarchical structures as Grosz et al. have demonstrated.

- Disambiguation with task-specific knowledge.

- System interaction. The user's perception in the virtual world is restricted by the camera view. Enabling the user to inquire about the situation of the virtual world, would allow to command robots more efficiently. In addition, the system should be able to ask the user questions when the system cannot proceed only with current knowledge.

Acknowledgements

We thank Erick Gallesio for a nice Scheme interpreter STk, and Alice Development Team at CMU for the great graphics system. Also, many thanks to Timothy John Baldwin for reviewing and proofreading.

References

[1] J. L. Austin. *How to Do Things with Words*. Oxford, 1960.

[2] Matthew J. Conway. *Alice: Easy-to-Learn 3D Scripting for Novices*. PhD thesis, University of Virginia, 1997. `http://www.alice.org/`.

[3] Erick Gallesio. *STk Reference Manual version 4.0.1*. Universite' de Nice - Sophia Antipolis Laboratoire I3S, 10 1999. `http://kaolin.unice.fr/STk/`.

[4] B. J. Grosz, A. K. Joshi, and S. Weinstrin. Providing a unified account of definite noun phrases in discourse. In *ACL Proceedings*, pages 44–49, 1983.

[5] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, July-September 1986.

[6] M. Kameyama. A property-sharing constraint in centering. In *ACL Proceedings*, pages 200–206, 1986.

[7] Y. Shinyama, T. Tokunaga, and H. Tanaka. Processing of 3-d spatial relations for virtual agents acting on natural language instructions. In *Proceedings of The Second Workshop on Intelligent Virtual Agents*, pages 67–78, 1999.

[8] T. Winograd. *Understanding Natural Language*. Academic Press, 1972.