Controlling Animated Agents in Natural Language

Kotaro Funakoshi
Department of Computer Science,
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro,
Tokyo 152-8552, Japan
koh@cl.cs.titech.ac.jp

Takenobu Tokugana
Department of Computer Science,
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro,
Tokyo 152-8552, Japan
take@cl.cs.titech.ac.jp

Abstract

This paper presents a prototype dialogue system, $\mathcal{K}3$, in which a user can instruct agents through speech input to manipulate various objects in a 3-D virtual world. In this paper, we focus on two distinctive features of the $\mathcal{K}3$ system: plan-based anaphora resolution and handling vagueness in spatial expressions. After an overview of the system architecture, each of these features is described. We also look at the future research agenda of this system.

1 Introduction

SHRDLU (**?**) can be considered as the most important natural language understanding system. Although SHRDLU was not "embodied", having had only a small stick to manipulate objects, it certainly had several features that a conversational agent is supposed to have. It had a great potential, and it was very promising for future research on natural language understanding.

Recently better technologies have become available in speech recognition and natural language processing. Major breakthroughs in the area of computer graphics have enabled us to generate complex, yet realistic 3-D animated agents or embodied life-like agents in a virtual environment. Researchers are now in a good position to go beyond SHRDLU by combining these technologies (**?**). This paper presents a conversational animated agent system, $\mathcal{K}3$.

Since all the actions carried out by an agent of the $\mathcal{K}3$ system are visible, we can evaluate the performance of the system by observing its animation. Visualizing the agents' actions yields many interesting issues from a cognitive science point of view; more complex processes are involved than those found in most conventional natural language understanding systems.

After sketching out the overview of the $\mathcal{K}3$ system in section 2, Two distinctive features of $\mathcal{K}3$ are discussed in section 3, and 4. Finally, section 5 concludes the paper and looks at future research agenda.

2 System Overview

A screen shot of $\mathcal{K}3$ is shown in Fig. 1. There are two agents and several objects in a virtual world. The current system accepts simple Japanese utterances with anaphoric and elliptical expressions, such as "Walk to the desk." and "Further". The size of the lexicon is about 100 words.

Figure 1: A screenshot of $\mathcal{K}3$

The architecture of the $\mathcal{K}3$ is illustrated in Fig. 2. system. The speech recognition module receives the user's speech input and generates a sequence of words. The syntactic/semantic analysis module analyzes the word sequence to extract a case frame. This module accepts ill-formed speech input including postposition omission, inversion, and self-correction. At this stage, not all case slots are necessarily filled, because of ellipses in the utterance. Even in cases where there is no ellipsis, instances of objects are not identified at this stage.

Resolving ellipses and anaphora, and identifying instances in the world are performed by the discourse analysis module. Anaphora resolution and instance identification are achieved by using plan-knowledge, which will be described in section 3.

The discourse analysis module extracts the user's goal as well and hands it over to the planning modules, which build a plan to generate the appropriate animation. In other words, the planning modules translate the user's goal into animation data. However, the properties of these two ends are very different and straightforward translation is rather difficult. The user's goal is represented in terms of symbols, while the animation data is a sequence of numeric values. To bridge this gap, we take a two-stage approach – macro- and micro-planning.

During the macro-planning, the planner needs to know the physical properties of objects, such as their size, location and so on. For example, to pick up a ball, the agent first needs to move to the location at which he can reach the ball. In this planning process, the distance between the ball and the agent needs to be calculated. This sort of information is represented in terms of coordinate values of the virtual space and handled by the micro-planner.

To interface the macro- and micro-planning, we introduced the SPACE object to represent a location in the virtual space by its symbolic and numeric character. The SPACE object is described in section 4.

3 Plan-based Anaphora Resolution

3.1 Surface-clue-based Resolution vs. Plan-based Resolution

Consider the following two dialogue examples.

(1-1) "Agent X, push the red ball."

(1-2) "Move to the front of the blue ball."

(1-3) "Push *it*."

(2-1) "Agent X, pick up the red ball."

(2-2) "Move to the front of the blue ball."

(2-3) "Put *it* down."

The second dialogue is different from the first one only in terms of the verbs in the first and third utterances. The syntactic structure of each sentence in the second dialogue (2-1)–(2-3) is the same as the corresponding sentence in the first dialogue (1-1)–(1-3). However, pronoun "it" in (1-3) refers to "the blue ball" in (1-2), and pronoun "it" in (2-3) refers to "the red ball" in (2-1). The difference between these two examples is not explained by the theories based on surface clues such as the centering theory (**?**; **?**).

In the setting of SHRDLU-like systems, the user has a certain goal of arranging objects in the world, and constructs a plan to achieve it through interaction with the system. As Cohen pointed out, users tend to break up the referring and predicating functions in speech dialogue (**?**). Thus, each user's utterance suggests a part of plan rather than a whole plan that the user tries to perform. To avoid redundancy, users need to use anaphora. From these observations, we found that considering a user's plan is indispensable in resolving anaphora in this type of dialogue system and developed an anaphora resolution algorithm using the relation between utterances in terms of partial plans (plan operators) corresponding to them.

The basic idea is to identify a chain of plan operators based on their effects and preconditions. Our method explained in the rest of this section finds preceding utterances sharing the same goal as the current utterance with respect to their corresponding plan operators as well as surface linguistic clues.

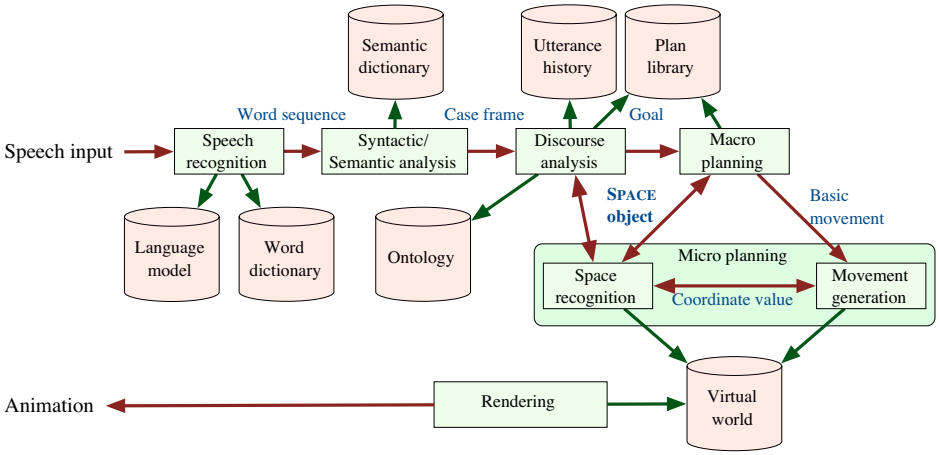

Figure 2: The system architecture of $\mathcal{K}3$

3.2 Resolution Algorithm

Recognized speech input is transformed into a case frame. At this stage, anaphora is not resolved. Based on this case frame, a plan operator is retrieved in the plan library. This process is generally called "plan recognition." A plan operator used in our system is similar to that of STRIPS (**?**), which consists of precondition, effect and action description.

Variables in the retrieved plan operator are filled with case fillers in the utterance. There might be missing case fillers when anaphora (zero pronoun) is used in the utterance. The system tries to resolve these missing elements in the plan operator. To resolve the missing elements, the system again uses clue words and the plan library. An overview of the anaphora resolution algorithm is shown in Figure 3.

When the utterance includes clue words, the system uses them to search the history database for the preceding utterance that shares the same goal as the current utterance. Then, it identifies the referent on the basis of case matching.

There are cases in which the proper preceding utterance cannot be identified even with the clue words. These cases are sent to the left branch in Fig. 3 where the plan library is used to resolve anaphora.

When there is no clue word or the clue word does not help to resolve the anaphora, the process goes through the left branch in Fig. 3. First, the system enumerates the candidates of referents using the surface information, then filters them out with linguistic clues and the plan library. For example, demonstratives such as "this", "that" are usually used for objects that are in the user's view. Therefore, the referent of anaphora with demonstratives is restricted to the objects in the current user's view.

If the effect of a plan operator satisfies the precondition of another plan operator, and the utterances corresponding to these plan operators are uttered in discourse, they can be considered to intend the same goal. Thus, identifying a chain of effect-precondition relations gives important information for grouping utterances sharing the same goal. We can assume an anaphor and its referent appear within the same utterance group.

Once the utterance group is identified, the system finds the referent based on matching variables between plan operators.

After filtering out the candidates, there still might be more than one candidate left. In such a case, each candidate is assigned a score that is calculated based on the following factors: saliency, agent's view, and user's view.

4 Handling Spatial Vagueness

To interface the macro- and micro-planning, we introduced the SPACE object which represents a location in the virtual world. Because of space limitations, we briefly explain the SPACE object.

The macro planner uses plan operators described in terms of the logical forms. Thus, the

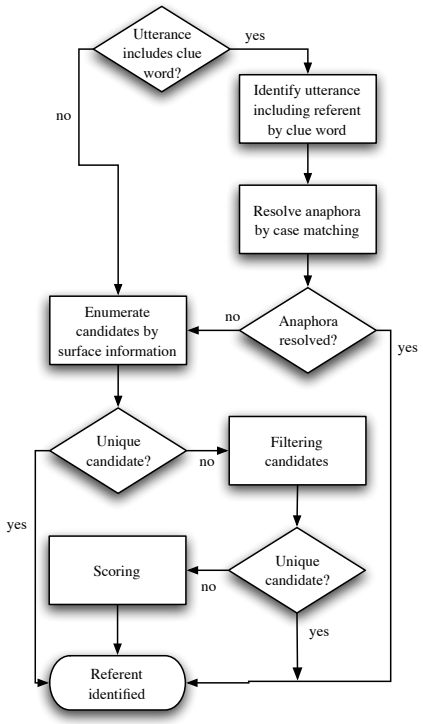

Figure 3: Anaphora resolution algorithm

SPACE object is designed to behave as a symbolic object in the macro-planning by referring to its unique identifier. On the other hand, a location could be vague and the most plausible place changes depending on the situation. Therefore, it should be treated as a certain region rather than a single point. To fulfill this requirement, we adopt the idea of the potential model proposed by Yamada et al. (**?**). Vagueness of a location is naturally realized as a potential function embedded in the SPACE object. The most plausible point is calculated by using the potential function with the Steepest Descent Method on request.

Consider the following short conversation between a human (H) and a virtual agent (A).

H: Do you see a ball in front of the desk?

A: Yes.

H: Put it on the desk.

When the first utterance is given in the situation shown in Fig. 1, the discourse analysis module identifies an instance of "a ball" in the following steps.

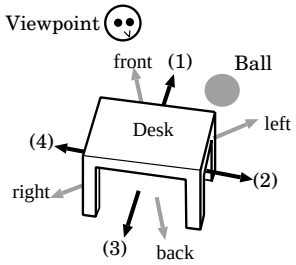

Figure 4: Adjustment of axis

(A) space#1 := new inFrontOf(desk#1, viewpoint#1, MIRROR)

(B) list#1 := space#1.findObjects()

(C) ball#1 := list#1.getFirstMatch(kindOf(BALL))

In step (A), an instance of SPACE is created as an instance of the class inFrontOf. The constructor of inFrontOf takes three arguments: the reference object, the viewpoint, and the axis order. Although it is necessary to identify the reference frame, we focus on the calculation of potential functions given a reference frame.

Suppose the parameters of inFrontOf have been resolved in the preceding steps, and the discourse analysis module chooses the axis mirror order and the orientation of the axis based on the viewpoint of the light-colored arrows in Fig. 4. The closest arrow to the viewpoint-based "front" axis ((1) in Fig. 4) is chosen as the "front" of the desk. Then, the parameters of potential function corresponding to "front" are set.

In step (B), the method matchObjects() returns a list of objects located in the potential field of space#1 shown in Fig. 5. The objects in the list are sorted in descending order of the potential value of their location.

In step (C), the most plausible object satisfying the type constraint (BALL) is selected by the method getFirstMatch().

When receiving the next utterance, "Put it on the desk.", the discourse analysis module resolves the referent of the pronoun "it" and extracts the user's goal.

```
walk(inFrontOf(ball#1, viewpoint#1, MIRROR)
AND reachableByHand(ball#1)
AND NOT(occupied(ball#1)))
```

The movement walk takes a SPACE object representing its destination as an argument. In this example, the conjunction of three SPACE objects

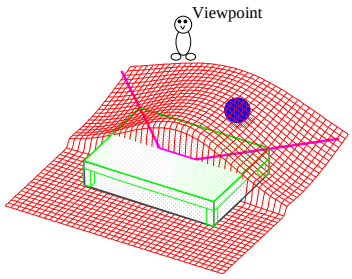

Figure 5: Potential field of space#1

is given as the argument. The potential function of the resultant SPACE is calculated by multiplying the values of the corresponding three potential functions at each point.

As this example illustrates, the SPACE object effectively plays a role as a mediator between the macro and micro planning.

5 Conclusions and Future Work

We have introduced our prototype system $\mathcal{K}3$, two distinctive features of which are described in this paper. Plan-based anaphora resolution enables $\mathcal{K}3$ to interpret the user's intention more precisely than the previous, surface-cue-based resolution algorithms. The SPACE object is designed to bridge the gap between the symbolic system (language processing) and the continuous system (animation generation) . In what follows, we describe the research agenda of our project.

One-to-many Conversation. Conversational agent systems should deal with one-to-many conversations as well as one-to-one conversations. In a one-to-many conversation, it is not easy to decide who is the intended listener. The situation gets worse when a speaker is concerned with only performing an action without caring who does it. In such cases, agents have to request clarifications or negotiate among themselves.

Agent Coordination. In one-to-many conversations, agents must coordinate each other. Some sorts of coordination are explicitly requested by user, *e.g.*, "Agent A and B tidy up the table, please." But other kinds of coordination are implicitly requested, *e.g.*, "Agent A hands agent B the box, please." In this case, the speaker asks agent B nothing explicitly. However, agent B must react to the request for agent A and coordinate with agent A to receive a box.

Parallel Actions. Most intelligent agent systems perform only one action at a time. Yet, if we want to make systems become more flexible, we must enable them to handle more than one action at a time. Hence, they must speak while walking, wave while nodding, and so on.

Memory System. A history database is not enough to serve realistic dialogue in the domain of $\mathcal{K}3$. In such a domain, people often mention a previous state, *e.g.*, "Put the ball back to the place where it was." To comply such a request, agents must have a human-like memory system.

Interruption Handling. Agents sometimes misunderstand requests and perform not intended actions. In case of human conversations, a speaker usually interrupts hearer and try to repair misunderstanding. Conversational agents also should be able to accept such interruptions. Interruption handling is also essential to request a next action before agents finish actions.

Acknowledgment

This work is partially supported by a Grant-in-Aid for Creative Scientific Research 13NP0301, the Ministry of Education, Culture, Sports, Science and Technology of Japan. The URL of the project is `http://www.cl.cs.titech.ac.jp/sinpro/en/index.html`.

References

P. R. Cohen. 1984. The pragmatics of referring and the modality of communication. *Computational Linguistics*, 10(2):97–146.

R. E. Fikes. 1971. STRIPS: A new approach to the application of theorem problem solving. *Artificial Intelligence*, 2:189–208.

B. J. Grosz, A. K. Joshi, and P. Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226.

H. Tanaka, T. Tokunaga, and Y. Shinyama. 2004. Animated agents capable of understanding natural language and performing actions. In *Life-Like Characters*, pages 429–444. Springer.

M. A. Walker, A. K. Joshi, and E. F. Prince, editors. 1998. *Centering Theory in Discourse*. Clarendon Press Oxford.

T. Winograd. 1972. *Understanding Natural Language*. Academic Press.

A. Yamada, T. Nishida, and S. Doshita. 1988. Figuring out most plausible interpretation from spatial description. In *the 12th International Conference on Computational Linguistics (COLING)*, pages 764–769.