$\mathcal{K}2$: Animated Agents that Understand Speech Commands and Perform Actions

Takenobu Tokugana, Kotaro Funakoshi, and Hozumi Tanaka

Department of Computer Science, Tokyo Institute of Technology
Tokyo Meguro Ôokayama 2-12-1, Japan
{take,koh,tanaka}@cl.cs.titech.ac.jp

Abstract. This paper presents a prototype dialogue system, $\mathcal{K}2$, in which a user can instruct agents through speech input to manipulate various objects in a 3-D virtual world. The agents' action is presented to the user as an animation. To build such a system, we have to deal with some of the deeper issues of natural language processing such as ellipsis and anaphora resolution, handling vagueness, and so on. In this paper, we focus on three distinctive features of the $\mathcal{K}2$ system: handling ill-formed speech input, 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

From a historical point of view, Winograd's SHRDLU [1] can be considered as the most important natural language understanding system. SHRDLU was a kind of software agent working in a block world. Although SHRDLU was not "embodied", having had only a small stick, it certainly had several features that a conversational agent is supposed to have. It could understand English through keyboard inputs and carry out some simple tasks such as "Pick up a red block on the table" by building a plan to achieve it. Furthermore, it could solve some of the anaphoric ambiguities in input sentences. In short, SHRDLU was clearly ahead of its time. 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 [2].

According to Cassell et al. [3], conversational skills consist not only in the ability to understand and produce language, but also in the ability to perform the corresponding body movements (facial expressions, the use of hands, etc.), intonations and tonal expressions. All of them have regulatory functions for the process of conversation. Cassell and her collaborators have developed REA, an embodied conversational agent endowed with social, linguistic, and psychological knowledge. While REA stresses the importance of non-verbal functions in conversations, this paper presents a conversational animated agent system, $\mathcal{K}2$, which emphasizes the importance of natural language understanding in spoken language. Although linguistic expressions handled by $\mathcal{K}2$ are limited, a number of issues remain to be addressed.

Since all the actions carried out by an agent of the $\mathcal{K}2$ 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. In this paper, we particularly focus on handling ill-formed speech input, resolving anaphora in the virtual world, handling vagueness in spatial expressions, and describe how the $\mathcal{K}2$ system approaches these issues.

After sketching out the overview of the $\mathcal{K}2$ system in Sect. 2, the above three issues are discussed in Sect. 3, 4, and 5. Finally, Sect. 6 concludes the paper and looks at future research agenda.

2 System Overview

A screen shot of $\mathcal{K}2$ is shown in Fig. 1. There are two agents and several objects (colored balls and desks) in a virtual world. Through speech input, a user can command the agents to manipulate the objects. 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. The agent's behavior and the subsequent changes in the virtual world are presented to the user in terms of a three-dimensional animation.

Fig. 1. A screenshot of $\mathcal{K}2$

The architecture of the $\mathcal{K}2$ is illustrates 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. Handling ill-formedness is described in Sect. 3. 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 Sect. 4.

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 Sect. 5.

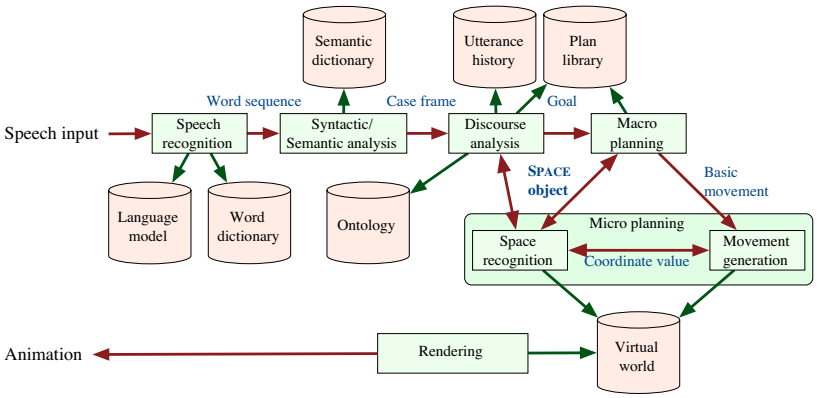

Fig. 2. The system architecture of $\mathcal{K}2$

3 Handling Ill-formed Speech Input

The syntactic/semantic analysis module in Fig. 2 adopts a phrase-based dependency parser in order to deal with spontaneous speech robustly. It handles the four types of ill-formed Japanese speech: postposition omission, inversion, self-correction, and hesitation. Here, we briefly describe the first three of them and how the parser deals with them. A more detailed description is found in [4].

Postposition Omission. In Japanese, the grammatical role of a noun phrase is marked by a postposition, and the order of postpositional phrases is relatively free. However,

speakers often omit postpositions, and this causes difficulties in syntactic and semantic analysis. In addition, when we use automatic speech recognizers (ASRs) in dialogue systems, we have to cope with the misrecognition of postpositions. Because their acoustic energy tends to be weak, postpositions tend to be misrecognized (often deleted) more than content words by ASRs. The parser estimates omitted or deleted postpositions from semantic constraints.

Inversion. Since Japanese is a head-final language, sentences usually end with a predicate. In speech dialogue, however, speakers sometimes add several phrases after the predicate. We consider such cases to be inversion, and assume that these post-predicate phrases depend on the predicate. The parser only allows phrases that come after a main predicate to depend on the preceding predicate.

Self-correction. Self-correction is also known as speech repair, or simply repair. In Japanese, self-correction can be combined with postposition omission and inversion:

akai tama-(wo) mae-(ni) osite migi-no yatu-wo
red ball-(ACC) front-(GOAL) push right-GEN one-ACC
(Push the right red ball forward)

In this example, the speaker corrected *akai tama-(wo)* (*wo* was omitted) by adding the inverted pronoun phrase, *migi-no yatu-wo*. The parser detects self-corrections by observing stacks in which the parser stores analysis hypotheses, and merges repaired phrases and repairing phrases while removing conflicting (that is, repaired) information and preserving information that resides only in the repaired phrases.

4 Plan-based Anaphora Resolution

4.1 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 [5–7].

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 [8]. 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 th 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.

4.2 Resolution Algorithm

As described in Sect. 2, speech input is recognized by the ASR and the recognized word sequence is syntactically and semantically analyzed, then 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." Currently the mapping from an utterance to a plan operator is done based on the verb in the utterance. When a verb is missing in the utterance, the system recovers the missing verb by using clue words and referring to the history database and the plan library.

A plan operator used in our system is similar to that of STRIPS [9], which consists of precondition, effect and action description. There are cases in which the missing verb can be recovered by referring to constraints on variables in the plan operator.

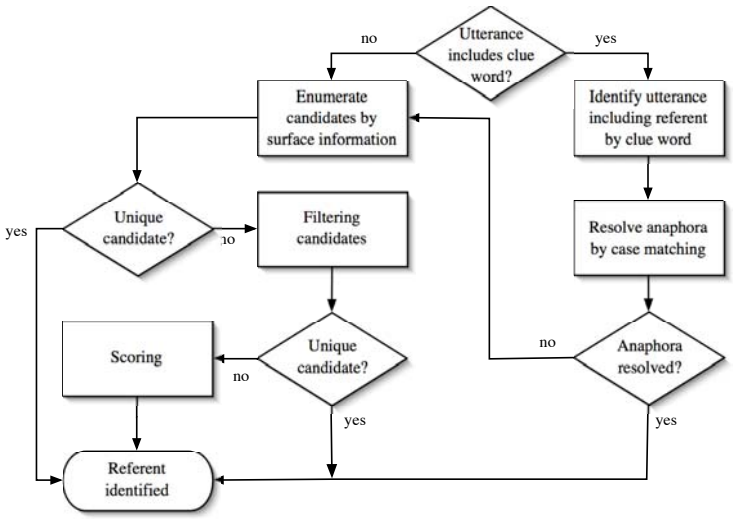

Fig. 3. Anaphora resolution algorithm

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.

5 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. Further details of the SPACE object are given in [10].

The macro planner uses plan operators described in terms of the logical forms, in which a location is described such as InFrontOf(Obj). Thus, the 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. [11], in which a potential function maps a location to its plausibility. Vagueness of a location is naturally realized as a potential function embedded in the SPACE object. When the most plausible point is required by the micro-planner for generating the animation, the point is calculated by using the potential function with the Steepest Descent Method.

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 an utterance "Do you see a ball in front of the desk?" is given in the situation shown in Fig. 1, the discourse analysis module identifies an instance of "a ball" in the following steps.

(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 [1]. Although it is necessary to identify the reference frame that the speaker used to interpret the speaker's utterance correctly, 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().

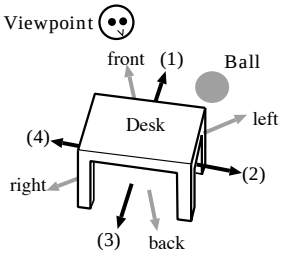

Fig. 4. Adjustment of axis

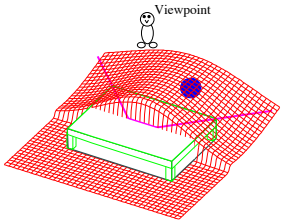

Fig. 5. Potential field of space#1

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. The macro planner constructs a plan to satisfy the goal as follows:

[1] We follow Herskovits' formulation [12] of spatial reference. There are two types of axis order: basic and mirror.

1. walk(inFrontOf(ball#1, viewpoint#1, MIRROR) AND reachableByHand(ball#1) AND NOT(occupied(ball#1)))
2. grasp(ball#1)
3. put(ball#1,on(desk#1, viewpoint#1, MIRROR))

Walk, grasp, and put are defined as basic movements. They are handed over to the micro planner one by one.

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

After moving to the specified location, the movement grasp is performed to grab ball#1. When putting the ball on the desk, the micro planner looks for a space on the desk that no other object occupies by composing the potential functions in a manner similar to the walk step.

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

6 Concluding Remarks and Future Work

We have introduced our prototype system $\mathcal{K}2$. $\mathcal{K}2$ has several distinctive features, three of which are described in this paper: handling ill-formed Japanese speech input, plan-based anaphora resolution, and handling spatial vagueness by bridging between macro- and micro-planning.

The system achieved robustness by introducing ill-formed input handling. Plan-based anaphora resolution enables $\mathcal{K}2$ 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), and it mediates between the two types of planners.

In what follows, we describe the research agenda of our project.

One-to-many Conversation. Natural language understanding systems should deal with not only face-to-face or one-to-one conversations, but also one-to-many conversations. One-to-many conversations typically take place in a multi-agent environment [13, 14]. In a one-to-one conversation, it is easy to decide who is the intended listener. In contrast, in a one-to-many conversation, there are many potential listeners, hence it should be decided at the beginning who is the intended listener. The intended listener is often mentioned explicitly in the early utterance of the dialogue, but this is not always the case. Without identifying the agent appointed as an actor of the action, a proper animation will not be generated. 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.

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.

Currently, the macro planner performs only a single action at a time, handing the micro planner the elements of each action one by one. To build a more versatile system, we have to develop a system able to carry out multiple actions at a time, simultaneously or sequentially, and we have to build an interface able to communicate between the macro- planner and the micro-planner.

Multimodality. In natural language understanding systems, multimodal information (gestures and gazing) is an important factor for interpreting a user's utterance. For example, pointing to a certain object could be an easy task if a pointing gesture is used together with an utterance. Obviously, this is what we are striving for: animated, natural looking agents.

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

1. Winograd, T.: Understanding Natural Language. Academic Press (1972)
2. Tanaka, H., Tokunaga, T., Shinyama, Y.: Animated agents capable of understanding natural language and performing actions. In: Life-Like Characters. Springer (2004) 429–444
3. Cassell, J., Bickmore, T., Billinghurst, L., Campbell, L., Chang, K., Vilhjalmsson, H., Yan, H.: Embodiment in conversational interfaces: REA. In: Proceedings of CHI'99 Conference. (1999) 520–527
4. Funakoshi, K., Tokunaga, T., Tanaka, H.: Processing Japanese self-correction in speech dialog systems. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING). (2002) 287–293
5. Grosz, B.J., Joshi, A.K.J., Weinstein, S.: Providing a unified account of definite noun phrases in discourse. In: Proceedings of ACL'83. (1983) 44–49
6. Grosz, B.J., Joshi, A.K., Weinstein, P.: Centering: A framework for modeling the local coherence of discourse. Computational Linguistics **21** (1995) 203–226
7. Walker, M.A., Joshi, A.K., Prince, E.F., eds.: Centering Theory in Discourse. Clarendon Press Oxford (1998)
8. Cohen, P.R.: The pragmatics of referring and the modality of communication. Computational Linguistics **10** (1984) 97–146
9. Fikes, R.E.: STRIPS: A new approach to the application of theorem problem solving. Artificial Intelligence **2** (1971) 189–208
10. Tokunaga, T., Koyama, T., Saito, S., Okumura, M.: Bridgin the gap between language and action. In: the 4th International Workshop on Intelligent Virual Agents. (2003) 127–135

11. Yamada, A., Nishida, T., Doshita, S.: Figuring out most plausible interpretation from spatial description. In: the 12th International Conference on Computational Linguistics (COLING). (1988) 764–769
12. Herskovits, A.: Language and Spatial Cognition. An Interdisciplinary Study of the Prepositions in English. Cambridge University Press (1986)
13. Ferber, J.: Multi-Agent Systems - An Introduction to Distributed Artificial Intelligence. Addison-Wesley Longman (1999)
14. Weiss, G., ed.: Multiagent Systems. The MIT Press (1999)