Taking Account of the User's View in 3D Multimodal Instruction Dialogue

Yukiko I. Nakano and **Kenji Imamura** and **Hisashi Ohara**
1-1 Hikari-no-oka, Yokosuka, Kanagawa, 239-0847 Japan
{yukiko, imamura, ohara}@nttnly.isl.ntt.co.jp

Abstract

While recent advancements in virtual reality technology have created a rich communication interface linking humans and computers, there has been little work on building dialogue systems for 3D virtual worlds. This paper proposes a method for altering the instruction dialogue to match the user's view in a virtual environment. We illustrate the method with the system MID-3D, which interactively instructs the user on dismantling some parts of a car. First, in order to change the content of the instruction dialogue to match the user's view, we extend the refinement-driven planning algorithm by using the user's view as a plan constraint. Second, to manage the dialogue smoothly, the system keeps track of the user's viewpoint as part of the dialogue state and uses this information for coping with interruptive subdialogues. These mechanisms enable MID-3D to set instruction dialogues in an incremental way; it takes account of the user's view even when it changes frequently.

1 Introduction

In a 3D virtual environment, we can freely walk through the virtual space and view three dimensional objects from various angles. A multimodal dialogue system for such a virtual environment should aim to realize conversations which are performed in the real world. It would also be very useful for education, where it is necessary to learn in near real-life situations.

One of the most significant characteristics of 3D virtual environments is that the user can select her/his own view from which to observe the virtual world. Thus, the multimodal instruction dialogue system should be able to set the course of the dialogue by considering the user's current view. However, previous works on multimodal presentation generation and instruction dialogue generation (Wahlster et al., 1993; Moore, 1995; Cawsey, 1992) do not achieve this goal because they were not designed to handle dialogues performed in 3D virtual environments.

This paper proposes a method that ensures that the course of the dialogue matches the user's view in the virtual environment. More specifically, we focus on (1) how to select the contents of the dialogue since it is essential that the instruction dialogue system form a sequence of dialogue contents that is coherent and comprehensible, and (2) how to control mixed-initiative instruction dialogues smoothly, especially how to manage interruptive subdialogues. These two problems basically determine the course of the dialogue.

First, in order to decide the appropriate content, we propose a content selection mechanism based on plan-based multimodal presentation generation (André and Rist, 1993; Wahlster et al., 1993). We extend this algorithm by using the user's view as a constraint in expanding the plan. In addition, by employing the incremental planning algorithm, the system can adjust the content to match the user's view during ongoing conversations.

Second, in order to manage interruptive subdialogues, we propose a dialogue management mechanism that takes account of the user's view. This mechanism maintains the user's viewpoint as a dialogue state in addition to intentional and linguistic context (Rich and Sidner, 1998). It maintains the dialogue state as a focus stack of discourse segments and updates it at each turn. Thus, it can track the viewpoint information in an on-going dialogue. By using this viewpoint information in resuming the dialogue after an interruptive subdialogue, the dialogue management mechanism returns the user's viewpoint to that of the interrupted segment.

These two mechanisms work as a core dialogue engine in MID-3D (Multimodal Instruction Dialogue system for 3D virtual environments). They make it possible to set the instruction dialogue in an incremental way while

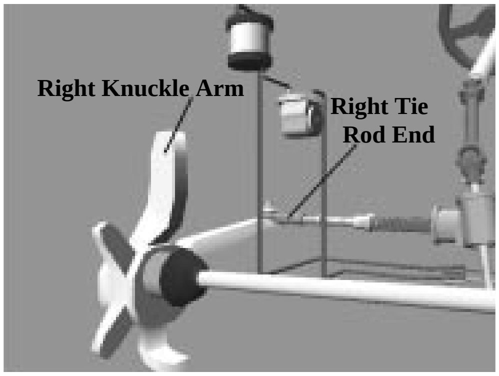

Figure 1: Right angle

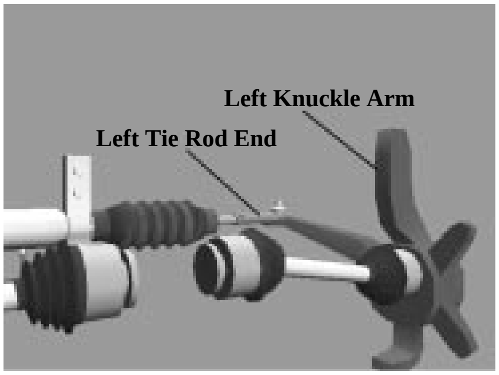

Figure 2: Left angle

considering the user's view. They also enable MID-3D to create coherent and mixed-initiative dialogues in virtual environments.

This paper is organized as follows. In Section 2, we define the problems specific to 3D multimodal dialogue generation. Section 3 describes related works. In Section 4, we propose the MID-3D architecture. Sections 5 and 6 describe the content planning mechanism and the dialogue management mechanism, and show they dynamically decide coherent instructions, and control mixed-initiative dialogues considering the user's view. We also show a sample dialogue in Section 7.

2 Problems

In a virtual environment, the user can freely move around the world and select her/his own view. The system cannot predict where the user will stand and what s/he observes in the virtual environment. This section describes two types of problems in generating instruction dialogues for such virtual environments. They are caused by mismatches between the user's viewpoint and the state of the dialogue.

First, the system should check whether the user's view matches the focus of the next exchange when the system tries to change communicative goals. If a mismatch occurs, the system should choose the instruction dialogue content according to the user's view. Figure 1 and 2 are examples of observing a car's front suspension from different points of view. In Figure 1, the right side of the steering system can be seen, while Figure 2 shows the left side. If the system is not aware of the user's view, the system may talk about the left tie rod end even though the user's view remains the right side (Figure 1). In such a case, the system should change its description or ask the user to change her/his view to the left side view (Figure 2) and recommence its instruction about this part. Therefore, the system should be able to change the content of the dialogue according to the user's view. In order to accomplish this, the system should have a content selection mechanism which incrementally decides the content while checking the user's current view.

Second, there could be a case in which the user changes the topic as well as the viewpoint as interrupting the system's instruction. In such a case, the dialogue system should keep track of the user's viewpoint as a part of the dialogue state and return to that viewpoint when resuming the dialogue after the interrupting subdialogue. Suppose that while the system is explaining the right tie rod end, the user initially looks at the right side (Figure 1) but then shifts her/his view to the left (Figure 2) and asks about the left knuckle arm. After finishing a subdialogue about this arm, the system tries to return to the dialogue about the interrupted topic. At this time, if the system resumed the dialogue using the current view (Figure 2), the view and the instruction would become mismatched. When resuming the interrupted dialogue, it would be less confusing to the user if the system returned to the user's prior viewpoint rather than selecting a new one. The user may be confused if the dialogue is resumed but the observed state looks different.

We address the above problems. In order to cope with the first problem, we present a content selection mechanism that incrementally expands the content plan of a multimodal dialogue while checking the user's view. To solve the second problem, we present a dialogue management mechanism that keeps track of the user's viewpoint as a part of the dialogue context and

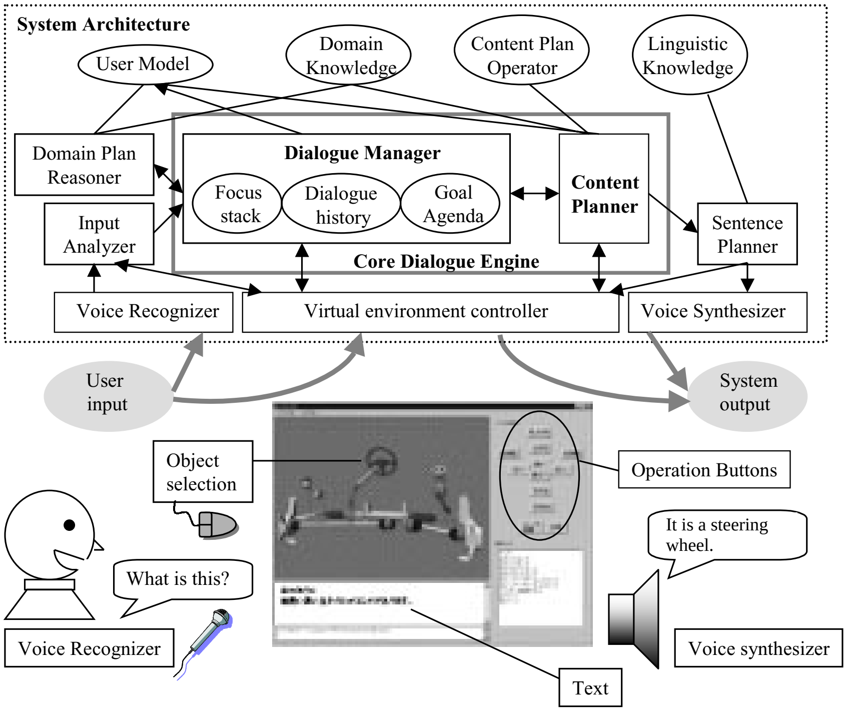

Figure 3: The system architecture

uses this information in resuming the dialogue after interruptive subdialogues.

3 Related work

There are many multimodal systems, such as multimedia presentation systems and animated agents (Maybury, 1993; Lester et al., 1997; Bares and Lester, 1997; Stone and Lester, 1996; Towns et al., 1998), all of which use 3D graphics and 3D animations. In some of them (Maybury, 1993; Wahlster et al., 1993; Towns et al., 1998), planning is used in generating multimodal presentations including graphics and animations. They are similar to MID-3D in that they use planning mechanisms in content planning. However, in presentation systems, unlike dialogue systems, the user just watches the presentation without changing her/his view. Therefore, these studies are not concerned with changing the content of the discourse to match the user's view.

In some studies of dialogue management (Rich and Sidner, 1998; Stent et al., 1999), the state of the dialogue is represented using Grosz and Sidner's framework (Grosz and Sidner, 1986). We also adopt this theory in our dialogue management mechanism. However, they do not keep track of the user's viewpoint information as a part of the dialogue state because they were not concerned with dialogue management in virtual environments.

Studies on pedagogical agents have goals closer to ours. In (Rickel and Johnson, 1999), a pedagogical agent demonstrates the sequential operation of complex machinery and answers some follow up questions from the student. Lester et al. (1999) proposes a lifelike pedagogical agent that supports problem-solving activities. Although these studies are concerned with building interactive learning environments using natural language, they do not discuss how to decide the course of on-going instruction dialogues in an incremental and coherent way.

4 Overview of the System Architecture

This section describes the architecture of MID-3D. This system instructs users how to dismantle the steering system of a car. The system steps through the procedure and the user can interrupt the system's instructions at any time. Figure 3 shows the architecture and a snapshot of the system. The 3D virtual environment is viewed through an application window. A 3D model of a part of the car is provided and a froglike character is used as the pedagogical agent (Johnson et al., 2000). The user herself/himself can also appear in the virtual environment as an avatar. The buttons to the right of the 3D screen are operation buttons for changing the viewpoint. By using these buttons, the user can freely change her/his viewpoint at any time.

This system consists of five main modules: Input Analyzer, Domain Plan Reasoner, Content Planner (CP), Sentence Planner, Dialogue Manager (DM), and Virtual Environment Controller.

First of all, the user's inputs are interpreted through the Input Analyzer. It receives strings of characters from the voice recognizer and the user's inputs from the Virtual Environment Controller. It interprets these inputs, transforms them into a semantic representation, and sends them to the DM.

The DM, working as a dialogue management mechanism, keeps track of the dialogue context including the user's view and decides the next goal (or action) of the system. Upon receiving an input from the user through the Input Analyzer, the DM sends it to the Domain Plan Reasoner (DPR) to get discourse goals for responding to the input. For example, if the user requests some instruction, the DPR decides the sequence of steps that realizes the procedure by referring to domain knowledge. The DM then adds the discourse goals to the goal agenda. If the user does not submit a new topic, the DM continues to expand the instruction plan by sending a goal in the goal agenda to the CP. Details of the DM are given in Section 6.

After the goal is sent to the CP, it decides the appropriate contents of instruction dialogue by employing a refinement-driven hierarchical linear planning technique. When it receives a goal from the DM, it expands the goal and returns its subgoal to the DM. By repeating this process, the dialogue contents are gradually specified. Therefore, the CP provides the scenario for the instruction based on the control provided by the DM. Details of the CP are provided in Section 5.

The Sentence Planner generates surface linguistic expressions coordinated with action (Kato et al., 1996). The linguistic expressions are output through a voice synthesizer. Actions are realized through the Virtual Environment Controller as 3D animation.

For the Virtual Environment Controller, we use HyCLASS (Kawanobe et al., 1998), which is a 3D simulation-based environment for educational activities. Several APIs are provided for controlling HyCLASS. By using these interfaces, the CP and the DM can discern the user's view and issue an action command in order to change the virtual environment. When HyCLASS receives an action command, it interprets the command and renders the 3D animation corresponding to the action in real time.

```
<Operator 1>
 (:Header          (Instruct-act S H ?act MM)
 :Effect           (BMB S H (Goal H (Done H ?act)))
 :Constraints      ((KB (Obj ?act ?object))
                   (Visible-p (Visible ?object t)))
 :Main-Acts        ((Look S H)
                    (Request S H (Try H (action ?act)) NO-SYNC MM))
 :Subsidiary-Acts  ((Describe- act S H ?act MM)
                    (Reset S (action ?act))))

<Operator 2>
 (:Header          (Instruct-act S H ?act MM)
 :Effect           (BMB S H (Goal H (Done H ?act)))
 :Constraints      ((KB (Obj ?act ?object))
                     (Visible-p (Visible ?object nil)))
 :Main-Acts        ((Look S H)
                     (Make-recognize S H (Object ?object) MM)
                     (Request S H (Try H (action ?act)) NO-SYNC MM))
 :Subsidiary-Acts  ((Describe-act S H ?act MM)
                     (Reset S (action ?act))))
```

Figure 4: Examples of Content Plan Operators

5 Selecting the Content of Instruction Dialogue

In this section, we introduce the CP and show how the instruction dialogue is decided in an incremental way to match the user's view.

5.1 Content Planner

In MID-3D, the CP is called by the DM. When a goal is put to the CP from the DM, it selects a plan operator for achieving the goal, applies the operator to find new subgoals, and returns them to the DM. The subgoals are then added to the goal agenda maintained by the DM. Therefore, the CP provides the scenario for the instruction dialogue to the DM and enables MID-3D to output coherent instructions. Moreover, the Content Planer employs depth-first search with a refinement-driven hierarchical linear planning algorithm as in (Cawsey, 1992). The advantage of this method is that the plan is developed incrementally, and can be changed while the conversation is in progress. Thus, by applying this algorithm to 3D dialogues, it becomes possible to set instruction dialogue strategies that are contingent on the user's view.

5.2 Considering the User's View in Content Selection

In order to decide the dialogue content according to the user's view, we extend the description of the content plan operator (André and Rist, 1993) by using the user's view as a constraint in plan operator selection. We also modify the constraint checking functions of the previous planning algorithm such that HyCLASS is queried about the state of the virtual environment.

Figure 4 shows examples of content plan operators. Each operator consists of the name of the operator (Header), the effect resulting from plan execution (Effect), the constraints for executing the plan (Constraints), the essential subgoals (Main-acts), and the optional subgoals (Subsidiary-acts). As shown in ⟨Operator 1⟩ in Figure 4, we use the constraint `(Visible-p (Visible ?object t))` to check whether the object is visible from the user's viewpoint. Actually, the CP asks HyCLASS to examine whether the object is in the student's field of view.

If an object is bound to the `?object` variable by referring to the knowledge base, and the object is visible to the user, ⟨Operator 1⟩ is selected. As a result, two Main-Acts (looking at the user and requesting to try to do the action) and two Subsidiary-Acts (showing how to do the action, then resetting the state) are set as subgoals and returned to the DM. In contrast, if the object is *not* visible to the user, ⟨Operator 2⟩ is selected. In this case, a goal for making the user identify the object is added to the Main-Acts; `(Make-recognize S H (Object ?object) MM)`.

As shown above, the user's view is considered in deciding the instruction strategy. In addition to the above example, the distance between the target object and the user as well as three dimensional overlapping of objects, can also be considered as constraints related to the user's view.

Although the user's view is also considered in selecting locative expressions of objects in the Sentence Planner in MID-3D, we do not discuss this issue here because surface generation is not the focus of this paper.

6 Managing Interruptive Subdialogue

The DM controls the other components of MID-3D based on a discourse model that represents the state of the dialogue. This section describes the DM and shows how the user's view is used in managing the instruction dialogue.

6.1 Maintaining the Discourse Model

The DM maintains a discourse model for tracking the state of the dialogue. The discourse model consists of the discourse goal agenda (agenda), focus stack, and dialogue history. The agenda is a list of goals that should be achieved through a dialogue between the user and the system. If all the goals in the agenda are accomplished, the instruction dialogue finishes successfully. The focus stack is a stack of discourse segment frames (DSF). Each DSF is a frame structure that stores the following information as slot values:

- *utterance content (UC):* A list of utterance contents constructing a discourse segment. Physical actions are also regarded as utterance contents (Ferguson and Allen, 1998).
- *discourse purpose (DP):* The purpose of a discourse segment.
- *goal state (GS):* A state (or states) which should be accomplished to achieve the discourse purpose of the segment.

In addition to these, we add the user's viewpoint slot to the DSF description in order to track the user's viewpoint information:

- *user's viewpoint (UV):* Current user's viewpoint, which is represented as the position and orientation of the camera. The position consists of x-, y-, and z-coordinates. The orientation consists of x-, y-, and z-angles of the camera.

The basic algorithm of the DM is to repeat (a) the performing actions step and (b) updating the discourse model, until there is no unsatisfied goal in the agenda (Traum, 1994). In performing actions step, the DM decides what to do next in the current dialogue state, and then performs the action. When continuing the system explanation, the DM posts the first goal in the agenda to the CP. If the user's response is needed in the current state, the DM waits for the user's input.

The other step in the DM algorithm is to update the discourse model according to the state that results from the actions performed by the user as well as the actions performed by the system. Although we do not detail this step here, the following operations could be executed depending on the case. If the current discourse purpose is accomplished, the top level DSF is popped and added to the dialogue history. The

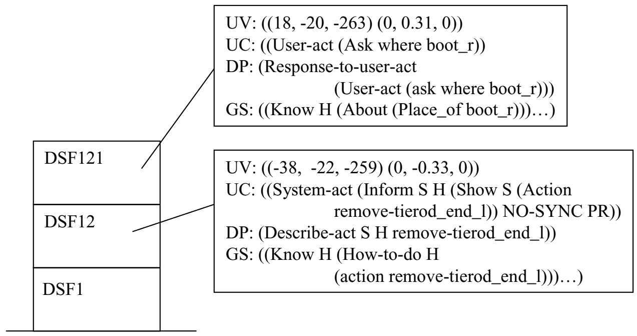
UV: ((18, -20, -263) (0, 0.31, 0))
UC: ((User-act (Ask where boot_r))
DP: (Response-to-user-act
(User-act (ask where boot_r)))
GS: ((Know H (About (Place_of boot_r)))…)
DSF121
DSF12
DSF1
UV: ((-38, -22, -259) (0, -0.33, 0))
UC: ((System-act (Inform S H (Show S (Action
remove-tierod_end_l)) NO-SYNC PR))
DP: (Describe-act S H remove-tierod_end_l))
GS: ((Know H (How-to-do H
(action remove-tierod_end_l)))…)

Figure 5: Example of the state of a dialogue

system then assumes that the user understands the instruction and adds the assumption to the user model. If a new discourse purpose is introduced from the CP, the DM creates a new DSF by setting the header of the selected plan operator in the discourse purpose slot and the effect of the operator in the goal state slot. The DSF is then pushed to the focus stack. If the current discourse purpose is continued, the DM updates the information of the top level DSF.

6.2 Considering the User's View in Coping with Interruptive Subdialogues

The main difference of the Dialogue Manager of our system from the previous one is to maintain the user's viewpoint information and use this in managing the dialogue. When the DM updates the information of the current DSF, it observes the user's viewpoint at that point and renews the UV slot and it also adds the semantic representation of utterance (or action) in the UC slot. As a result, it becomes possible to update the user's viewpoint information at each turn, and to track the user's viewpoint in an on-going dialogue.

By using this mechanism, the DM can cope with interruptive subdialogues. In resuming from a subdialogue, the user may become confused if the dialogue is resumed but the observed state differs from what the user remembers. In order to match the view to the resumed dialogue, the DM refers the UV slot of the top DSF and puts the users view back to that point. This ensures that the user experiences a smooth transition back to the previous topic. Figure 5 shows an example of the state of a dialogue. DSF12 represents a discourse segment that describes how to remove the left tie rod end. DSF121 represents the user-initiated interruptive subdialogue about where the right boot is. Immediately before starting DSF121, the user's viewpoint in DSF12 is ((-38, -22, -259) (0, -0.33, 0)). After completing the subdialogue by answering the user's question, DSF121 is popped and the system resumes DSF12. At this time, the DM gets the viewpoint value of the top DSF (DSF12), and commands HyCLASS to change the viewpoint to that view, which is in this case ((-38, -22, -259) (0, -0.33, 0)) [1] . The system then restarts the interrupted dialogue.

[14]System: The left knuckle arm is removed like this.
(with the animation showing the left knuckle arm coming off)
[15]User: (After moving the viewpoint to Figure1 and clicking the right knuckle arm)
What is this?
[16]System: This is the right knuckle arm.
[17]User: OK.
[18]System: Now, let's continue the explanation.
(with changing the view to the one in utterance [14])
[19]System: The left knuckle arm is removed like this.
(with the animation showing the left knuckle arm coming off)
[20]User: Yes.
[21]System: Now, try removing the left knuckle arm.
[22]User: (After clicking the left knuckle arm) Remove it.
[23]System: You have successfully completed the operation.
[24]System: The next step is to remove the right tie rod end.
[25]System: Let's move to the right tie rod end.
[26] User: Yes.
[27]System: (Moving the view to the right)
The right tie rod end is in front of you.
[28] User: Yes, I see.

Figure 6: Example of a dialogue with MID-3D

7 Example

In order to illustrate the behavior of MID-3D, an example is shown in Figure 6. This is a part of an instruction dialogue on how to dismantle the steering system of a car. The current topic is removing the left knuckle arm. In utterance [14], the system describes how to remove this part in conjunction with an animation created by HyCLASS.

In [15], the user interrupted the system's instruction and asked "What is this?" by clicking the right knuckle arm. At this point, the user's speech input was interpreted in the Input An-

[1] In the current system, it is not possible to move the camera to an arbitrary point because of the limitations of the virtual environment controller employed. Accordingly, this function is approximated by selecting the nearest of several predefined viewpoints.

alyzer and a user initiative subdialogue started by pushing another DSF onto the focus stack. In order to answer the question, the DM asked the Domain Plan Reasoner how to answer the user's question. As a result, a discourse goal was returned to the DM and added to the agenda. The DM then sent the goal `(Describe-name S H (object knuckle_arm_r))` to the CP. This goal generated utterance [16].

In system utterance [18], in order to resume the dialogue, a meta-comment, "Now let's continue the explanation", was generated and the viewpoint returned to the previous one in [14] as noted in the DSF. After returning to the previous view, the interrupted goal was re-planned. As a result, utterance [19] was generated.

After completing this operation in [23], the next step, removing the right tie rod end, is started. At this time, if the user is viewing the left side (Figure 2) and the system has the goal `(Instruct-act S H remove-tierod_end_r MM)`, ⟨Operator 2⟩ in Figure 4 is applied because the target object, right tie rod end, is not visible from the user's viewpoint. Thus a goal of making the user view the right tie rod end is added as a subgoal and utterances [24] and [25] are generated.

8 Discussion

This paper proposed a method for altering instruction dialogues to match the user's view in a virtual environment. We described the Content Planner which can incrementally decide coherent instruction dialogue content to match changes in the user's view. We also presented the Dialogue Manager, which can keep track of the user's viewpoint in an on-going dialogue and use this information in resuming from interruptive subdialogues. These mechanisms allow to detect mismatches between the user's viewpoint and the topic at any point in the dialogue, and then to choose the instruction content and user's viewpoint appropriately. MID-3D, an experimental system that uses these mechanisms, shows that the method we proposed is effective in realizing instruction dialogues that suit the user's view in virtual environments.

References

Elisabeth André and Thomas Rist. 1993. The design of illustrated documents as a planning task. In Mark T. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 94–116. AAAI Press / The MIT Press.

William H. Bares and James C. Lester. 1997. Realtime generation of customized 3D animated explanations for knowledge-based learning environments. In *AAAI97*, pages 347–354.

Alison Cawsey. 1992. *Explanation and Interaction: The Computer Generation of Expalanatory Dialogues.* The MIT Press.

George Ferguson and James F. Allen. 1998. TRIPS: An integrated intelligent problem-solving assistant. In *AAAI98*, pages 567–572.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

W. Lewis Johnson, Jeff W. Rickel, and James C. Lester. 2000. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education.*

Tsuneaki Kato, Yukiko I. Nakano, Hideharu Nakajima, and Takaaki Hasegawa. 1996. Interactive multimodal explanations and their temporal coordination. In *ECAI-96*, pages 261–265. John Willey and Sons Limited.

Akihisa Kawanobe, Susumu Kakuta, Hirofumi Touhei, and Katsumi Hosoya. 1998. Preliminary report on HyCLASS authoring tool. In *ED-MEDIA/ED-TELECOM.*

James C. Lester, Jennifer L. Voerman, Stuart G. Towns, and Charles B. Callaway. 1997. Cosmo: A life-like animated pedagogical agent with deictic believability. In *IJCAI-97 Workshop, Animated Interface Agent.*

James C. Lester, Brian A. Stone, and Gray D. Stelling. 1999. Lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments. *User Modeling and User-Adapted Interaction*, 9(1-2):1–44.

Mark T. Maybury. 1993. Planning multimedia explanation using communicative acts. In Mark T. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 59–74. AAAI Press / The MIT Press.

Johanna D. Moore. 1995. *Participating in Explanatory Dialogues: Interpreting and Responding to Questions in Context.* MIT Press.

Charles Rich and Candace L. Sidner. 1998. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8:315–350.

Jeff W. Rickel and W. Lewis Johnson. 1999. Animated agents for procedual training in virtual reality: Perception, cognition and motor control. *Applied Artificial Intellifence*, 13:343–392.

Amanda Stent, John Dowding, Jean Mark Gawron, Elizabeth Owen Brat, and Robert Moore. 1999. The CommandTalk spoken dialogue system. In *ACL99*, pages 183–190.

Brian A. Stone and James C. Lester. 1996. Dynamically sequencing an animated pedagogical agent. In *AAAI96*, pages 424–431.

Stuart G. Towns, Charles B. Callaway, and James C. Lester. 1998. Generating coordinated natural language and 3D animations for complex spatial explanations. In *AAAI98*, pages 112–119.

David R. Traum. 1994. *A Computational Theory of Grounding in Natural Language Conversation.* Ph.D. thesis, University of Rochester.

Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, and Thomas Rist. 1993. Plan-based integration of natural language and graphics generation. *Artificial Intelligence*, 63:387–427.