12 31

# Discourse Style and Situation Viewpoint for a Conversational Language Tutor

Raza Hashim and Henry Hamburger
Department of Computer Science
George Mason University
Fairfax, Va 22030
email:rhashim@gmuvax2.gmu.edu
henryh@aic.gmu.edu

## Perspective, Rationale and Organization

hat should a conversational foreign language tutoring system say next? The choice is nstrained by two sets of considerations: the demands of conversational continuity and the iguage progress of the student. An introduction to these potentially conflicting quirements is the topic of an earlier paper [1]. The present study provides a more hnical analysis of the conversational side of the problem by developing a framework with o key conversational structures: interaction mode and conversational viewpoint. In the urse of the investigation, it also becomes necessary to examine issues of knowledge presentation and implementation for a two-way, two-medium communication system, one at lets tutor and student converse spatially and linguistically.

t the outset, the reader may well wonder just what a conversational language tutoring stem might look like and why one should expect it to be useful. One may also seek some ue to how it is organized, to gain perspective on the role played by discourse and ewpoint, the main subject matter of this paper. We take up these preliminary questions iefly, in turn.

e purpose of a conversational - or immersive - language learning environment is to help e student to understand and use a new foreign language directly and automatically, as

opposed to carrying out conscious mental translation or computation of grammatical formulas. To this end, we get the computer to provide some of the immersion activities that successful language tutors often use, involving familiar physical objects used to act out familiar scenarios. Following these methods, we make use of the student's own (first) language only to set the stage, and leave the explicit teaching of grammar to others. Immersion must be gradual, with exposure and comprehension of language preceding its production. Gradualness also means introducing new words and patterns one at a time, in situations where they can be figured out from the rest of the sentence in the context of the visual activity and the continuity of the subject matter in the conversation. It is this need for conversational continuity that motivates the work below.

To fix ideas, it may help to have a concrete example of the kind of objects and scenarios that are involved here. In one microworld that we call Washroom World, there is a figure whose movable hand enables him to turn on the water and to pick up, use and put down various objects like the soap, towel or toothbrush as needed. Such actions can combine to form composite actions that result in getting his face or teeth clean, and so on. The hand can be controlled by either student or system, and the sequence of events is quite flexible. Foreign language descriptions, comments, commands and questions are tightly interwoven with the screen events.

Space does not permit proper treatment of the second and third preliminary questions, those of motivation and organization. On motivation, see our earlier work [2] and the broader discussion by [3]. The approach is best suited for the early stages and can be used with other systems - as well as teachers and books - that complement it.

As for organization, viewing matters very coarsely, one may say that the system has three broad functions: to handle input, make internal updates and decisions, and generate output. Breaking these down in turn, input involves handling what the student says and does. This means robust natural language understanding with error analysis as well as interpretation of graphics (mouse) actions, each interpreted in their mutual context. As for output, it too is bimodal and involves corresponding challenges. Internally, the system is responsible for naintaining models of the situation, the discourse, and the student, and for making tutorial lecisions about what to say and do. The preceding few sentences suggest ten modules, five ach for the internal duties and for input/output. Each of the ten offers ample opportunity or complex reasoning. To make such an ambitious undertaking manageable we are making ome judicious compromises and proceeding in stages. A pilot system has been built that provides some useful immersion experiences. A more modular, flexible and elaborate system is under construction.

## 2. The Structure of What to Say

The system and the student need to maintain a coherent dialog so that both parties have a physical and linguistic context in which to interpret and formulate new sentences. A concrete physical situation is important, since the dialog is in the new language, which the student doesn't understand completely. The partially animated graphical interaction on the computer screen indicates the physical context in which both the system and the student formulate and interpret utterances. The linguistic context, on the other hand, gradually develops as both parties succeed in communicating their intent to each other. This requirement of maintaining a shared context is present for conversation in general [4] but is more important for language tutors since they need a language-independent source of information about what is being said, to support the learning of new language aspects without translation.

Section 2.1 introduces our approach to the problem of conversational continuity and states some principles for achieving clarity in the conversation. The next section deals with microworlds, discussing what aspects need to be represented, including goals and actions. In section 2.3, we present dialog schemas and interaction types that have been developed to maintain conversational continuity. The final subsection indicates how language diversity can be achieved by alternative views of an event and its results.

### 2.1 Approach and Principles

To achieve conversational continuity the FLUENT system needs to (1) structure th student-tutor discourse, and (2) decide which aspect of the current situation to talk abou We structure the student-tutor discourse by maintaining a three-level discou representation. At the top level is the dialog schema, a skeletal plan of the student-tut interaction. A dialog schema is composed a flexible sequence of interchanges between th student and the tutor. Each interchange, in turn, comprises of a small number of turn possibly as few as one, by each party. Each turn is a linguistic and/or spatial output b either the student or the tutor. The second issue, what particular aspect or view of th situation to talk about, is the responsibility of the view selector, which must be sensitive bot to student needs and to the discourse structure just sketched. In the process of constructin the student-tutor dialog we adhere to the following principles that help maintain clarity.

For educational continuity:

1. Present input to the student in order of increasing linguistic complexity.
2. Present only one new linguistic aspect in a sentence and only a few in a lesson.
3. Assume comprehension of an aspect before demanding its production.

For conversational continuity:

4. Keep students aware of the discourse structure.
5. Make comments that are relevant to the current situation.
6. In a new situation, make view selection relatively repetitive.

To control the ambiguity of the visual channel:

7. Keep the number of objects low.
8. Talk about physical objects and their visible properties before talking about abstract o invisible properties.
9. Also defer talk of abstract actions and goals.

' Representing the Domain: Goals and Actions

this section we describe how to represent goals and actions in the microworld. Examples cribed in this section will be used in sections 2.3 and 2.4 to show how to talk about these ds and actions. The state information of the objects in the microworld has been resented as a simple object system with inheritance. Since our use of this approach to tes is familiar and straightforward, we focus on goals and actions.

a microworld, like the Washroom World mentioned above, we can talk about the goal that being currently pursued, such as washing the face, the action that is being carried out, h as picking up an object from the cabinet, or the state of a particular object, such as the e being dirty. In order to communicate effectively at these three levels, we maintain three els of domain representation. At the top level is a model of the goals that can be pursued the microworld. In the Washroom World example mentioned earlier, these goals include shing one's face, combing hair, brushing teeth, etc. With each goal there is a goal icture that tells the system how to achieve that goal, using primitive actions and/or other b)goals.

 action rules form the second level of representation in the system. There are rules for ions such as picking up and putting down objects, as well as manipulating the states of itches such as the faucets and the light switch. At the third level is a frame-based resentation of the underlying microworld, which keeps track of the visible properties of objects in the microworld, like the cup being on the shelf, as well as invisible properties, the surface state of the hand being wet.

pecific example of a goal structure is in Figure 1, which consists entirely of executable olog code. The goal in this example is to get a movable object - say a cup or a plate - py, using a sponge. The object to be soaped must meet the conditions in the "pre" list, ich indicates the the object is to be held in the stationary hand and should be dirty. The b" indicates how soaping is actually carried out, as a sequence of familiar actions. The roworld has a cabinet, a sink, a sponge, and a bucket of soapy water. The character has

a stationary hand, which can hold one item at a time, and a mobile hand, which can pick up one item at a time.

Two points are worth mentioning about this goal structure. First, it does not recognize an arbitrary sequence of actions as achieving a goal. A system constructed in this way will only recognize its predefined goal structures. In order to recognize arbitrary sequences as goals we would need a deeper model of the domain. Second, it is possible to define the goal, say soap(X), as that of getting the surface-condition slot of the object X as soapy using the predefined action rules of the system. We have chosen not to do this since planning a sequence of actions in the microworld to achieve a particular goal can be a nontrivial task. For details on microworld planning see [5].

The second level of representation correspond to actions that can be observed graphically. These actions include things such as pick-up, put-down, brush-teeth, soap-by-sponge, etc. Figure 2, which also is executable Prolog code, shows a typical action in the system. Since the character in the microworld has only one mobile hand, we have chosen to represent it by the constant, "m_hand". The rule transfers an object to the mobile hand from the thing that previously contained it.

The preconditions show that for this rule to be used, the mobile hand must be empty, the object to be transferred must be movable, and it must be located at the "From" object. In case the preconditions are met, the system can execute the rule. To do so, it issues the corresponding "act" and "mouse" commands to the graphics component to carry out the action graphically, and updates the knowledge base by carrying out all the items in the "post" or postconditions section of the rule.

The system can also use this rule in reverse to detect a pick-up action that is carried out by the student graphically. In order to do this it makes sure that the student's graphics actions matched the "mouse" and "act" component of this rule, and that all the preconditions are met. If this is the case, the system can update the knowledge base accordingly by asserting the postconditions.

```
goal_is     soap_by_sponge(Object)

pre         is_movable(Object) and
            is_dirty(Object) and
            in(s_hand,Object)

sub         action:pick_up(sponge,cabinet) and
            action:soap_sponge(sponge,soap_water_bucket) and
            action:soap_by_sponge(Object) and
            action:put_down(sponge,cabinet) and
            action:put_down(Object,sink)

post        Query =.. [Object, surface, =, [soapy]] and
            call(Query).
```

Figure 1: Goal Structure for Soaping a Movable Object.

1. The code given in this figure is executable Prolog code. Variable names begin with upper case letters while constants are lower case.

2. "goal_is", "pre", "sub" and "post" are user defined operators.

Choosing the level of detail is an important issue in the design of a microworld. The primitive graphic actions in our microworld are visit and mouse actions. It is possible to describe all actions in the system as a combination of these primitives. This, however, would mean reasoning from first principles about the state of the world, complicating the tasks of both microworld reasoning and language processing. We therefore model all actions with rules like the one in figure 2, expressing a higher level action in terms of graphic primitives and microworld states. Such rules represent complex actions that are applicable in rather special circumstances. Beside the one shown, other complex, specific rules deal with actions like wetting movable objects, soaping objects by hand when the surface of the hand is soapy, and so on. For interpreting student actions, these rules are preferred over the broadly applicable primitives. Modeling actions at this level simplifies language processing as well as reasoning, since languages typically have verbs at this level of abstraction.

```
rule        pick_object

do          goal(pick_up(Object,From))

mouse       mouse(mouseDown)

act         visit(m_hand,Object)

pre         value(m_hand, contains, =, empty) and
            value(Object,is_prop, =,Props) and
            member(is_movable,Props) and
            value(From,contains, =, Object)

post        Fact =.. [From,contains, =,Object] and
            retract(Fact) and
            assert(hand(contains, =,Object)).
```

Figure 2: Rule to Pick-up an Object.

1. The code given in this figure is executable Prolog code. Variable names begin with up case letters while constants are lower case.

2. "rule", "do", "mouse", "act", "pre" and "post" are user defined operators wi appropriate precedence levels.

## 2.3 Dialog Schemas and Interaction Modes

In the previous section we looked at the knowledge representation needed to carry o various goals and actions in an immersion style microworld. This section covers the dial schemas, interaction types, and views which are discourse tools needed to talk about goal actions and states. We begin with dialog schemas, which guide communication with th student, and more specifically with the example in figure 3.

At the heart of the dialog schema in figure 3 are two types of interactions. In th presentation mode, the system simply describes a particular view of the current activity an the student simply acknowledges this presentation, while in the commander mode the tuto issues a command and the student carries it out graphically. In order to simplify th esentation of this example, it is assumed that the student will perform each graphic action ccessfully without errors in the commander mode. In the actual situation this might not the case. We can use discourse management strategies to make local plan repairs in case incorrect student responses and ensure that the student remains on track, for details see ].

```
.presentation(goal:Goal, agent:Agent,
                              view:[describe_goal_to_do])

 presentation(object:Object,slot:Slot, value:Value,
                              view:[describe_state_value])

  commander(goal:Goal,
                              view:[describe_action]).

 presentation(object:Object,slot:Slot, value:Value,
                              view:[describe_state_value_now]).

  presentation(object:Object,slot:Slot,value:OldValue,
                              view:[describe_state_value_before]).

  presentation(goal:Goal,
                              view:[describe_goal_action]).
```

Figure 3: A dialog schema that uses Presentation and Commander mode

uppose we instantiate this schema with the goal soap-by-sponge, with the object "glass" and lot "surface", and agent as the student. This can result in the bimodal dialog of figure 4. he sentences in the figure are to be produced by a natural language generation generator perating on the output representation produced by our system. We are collaborating with he Athena Language Learning Project [7] whose generator is capable of this kind of output.

 few specific comments may help with understanding figure 4. Items (3-7) correspond to he commander mode interaction while the other utterances correspond to the presentation node. We will look at the commander mode interactions (3-7) first. In these pairs the utorial command is followed by the student performing the appropriate graphic actions in esponse. The expression "visit(X,Y)" is generated when a rectangular frame for object X verlaps one for object Y. In item (7) student actions must trigger both pick-up and

put-down to take the glass from the stationary hand and put it in the sink. As a result of each tutorial command and student response, several updates occur in the microworld, and several tutorial comments become possible, by using different views, as explained in section 2.4.

| Tutor | Student |
|---|---|
| 1. You will soap the glass. | <acknowledge> |
| 2. It is dirty now. | <acknowledge> |
| 3. Pick up the sponge. | visit(m_hand,sponge), mouse(mouseDown) |
| 4. Soap it in water. | visit(sponge,soap_water) |
| 5. Take the sponge to the glass. | visit(sponge,glass) |
| 6. Put it back in the cabinet. | visit(sponge,cabinet), mouse(mouseDown) |
| 7. Put the glass down in the sink. | visit(m_hand,glass),mouse(mouseDown) visit(glass,sink), mouse(mouseDown) |
| 8. Now the glass is soapy. | <acknowledge> |
| 9. It was dirty, before. | <acknowledge> |
| 10. You soaped it with a sponge. | <acknowledge> |

Figure 4: A sample dialog based on the schema

If we add more interaction types more interesting discourse schemas can be created. Some of these occur in the current pilot system but are not implemented with the kind of flexibility and generality we ultimately seek. Quizmaster is defined as a type of interaction that allows the tutor to ask students questions which they can respond to graphically or linguistically. The Oracle interaction would let the student ask questions, either picking from a menu of options, or by typing in questions, and the system responds. In the Movecaster mode, which is currently implemented with some generality, the system comments on the graphic actions done by the student. Celebrity would be the reverse of Movecaster, allowing the student to talk about system moves; the system would check for appropriateness. In the Servant mode the student issues the commands and the tutor carries them out, again with the possibility of commenting on the goal, action and states that are relevant. Some of these interaction types and some views are less demanding of student progress than others, and the system designers can choose to hand-craft dialog schemas of increasing difficulty using these ideas.

## '.4 Views

l'here is a great diversity in what the tutor can say at various times. For one thing, the lialog schema given in Figure 3 can be used with other goal structures, for example, tructures for washing or drying objects. In this section we will see that even at one articular time there are many conversationally relevant utterances that can be produced by he language tutor. For example, after the interaction in line (3) of Figure 4, any of the ollowing sentences is reasonable:

|. The sponge is in the hand.

'. You picked up the sponge from the cabinet.

}. The sponge is no longer in the cabinet.

|. You are holding the sponge.

:ach of these utterances corresponds to a different view of the resulting situation. Such itterances can be generated by the tutor with more advanced students. With beginning tudents, a good tutorial strategy would be to stick to the basic command and very little ollow up. In case the student doesn't carry out the anticipated action, it would seem useful or the system to generate an utterance describing what the student actually did. Figure 5 ives a selection from among dozens of possible views involving goals, actions, time and he state of objects. Commenting on less obvious aspects of the situation can provide mportant language exposure to the student. However, some of these views might be lifficult to understand for the beginner since they correspond to the physical situation only n an indirect way.

\ view structure is essentially a frame, implemented in the same Prolog style as the action ules and goals. One slot contains the interaction type and view information. Another slot pecifies what information about the current situation will need to be consulted, typically the ind of action and its current (actual) arguments. A third slot specifies how to put the formation from the first two together to construct a semantic structure to send to the atural language generator. It should be noted that the dialog schema given in Figure 3 can

be used with other goal structures, for example, structures for washing or drying objects. Moreover, looking at the list of views in Figure 5 one can see that if they are incorporated into the dialog schema given in Figure 3 many more conversationally relevant utterances can be produced by the language tutor.

Goal-Oriented Views:
- starting subgoal
- subgoal transition within the same goal
- goal completion
- violation of goal-based expectations

Action-Oriented Views:
- action that was carried out
- non-occurrence of anticipated action
- same operator with another argument, negated

Temporal View:
- this action follows its predecessor

State-Oriented Views:
- new state of the object that has been acted upon
- previous state of the object acted upon
- composite observation ("Now, there are two ...")
- preceding state no longer holds

Figure 5: Views possible in a goal-oriented microworld

## 3 Summary

Discourse style and situational viewpoint are key elements of an immersion-style language tutor. In order to achieve conversational continuity, such a system needs to maintain a model of goals, actions and states in the microworld along with tools to manage discourse. The discourse management tools include various dialog schemas composed of interaction types. These schemas guide the communication protocol between the student and the tutor. Another tool is the type of view that one takes of what occurs in the microworld. The various view types in our taxonomy provide a mechanism for generating comments on those aspects of the microworld that are relevant while using a particular interaction type. In sum, we have devised a variety of interaction types and views and shown how to use them in the construction of a bimodal, bidirectional communication system for immersive foreign language learning.

## References

1. Hamburger, H. and Maney, T. (1991) Twofold Continuity in Immersive Language Learning. J. Computer-Assisted Language Learning 4,2: 81-92.

2. Hamburger, H. and Hashim, R. (1992) Foreign langauge tutoring and learning system. In M. Swartz and M. Yazdani (Eds.), Intelligent Tutoring Systems for Foreign Language Learning. New York: Springer-Verlag.

3. Richards, J.C. and Rodgers, T.S.(1986) Approaches and Methods in Language Teaching: A Description and Analysis. Cambridge: Cambridge Language Teaching Library.

4. Grosz, B. and Sidner, C. (1987) Attention, intentions, and structure of discourse. Computational Linguistics 12,3: 175-204.

5. Wilensky, R. (1983) Planning and Understanding: A Computational Approach to Human Reasoning. Reading, MA: Addison Wesley.

6. Novick, D. (1988) Doctoral Dissertation, University of Oregon, Eugene, Oregon.

7. Felshin, S. (1991) The Lingo Manual. The Athena Language Learning Project, MIT.