10 Interactivity in Cooperative Problem Solving With Computers

Gellof Kanselaar
Gijsbert Erkens
University of Utrecht

Research within the field of intelligent computer-assisted instruction (CAI) is focused mainly on domain-specific questions of content representation, student modeling, and didactic intervention by the program acting as a tutor. However, tutoring is interactive by nature. Its effect greatly depends on the coordination and fine tuning of communication between tutor and student. This process not only concerns the conceptual aspects of information exchange, but also involves knowledge about the communicative aspects of the specific problem-solving situation. Coordination of communication between student and system becomes even more crucial when their interactional roles are less asymmetrically defined and not as well divided as that of tutor and tutee. This is the case in educational systems, which are meant to adopt a more cooperative role toward the student, acting as a partner in dealing with a problem-solving task. In a cooperative situation, more symmetry, flexibility, and mixed initiative are required in the interaction between student and program. This can only be achieved by a mutually controlled process of coordination at a communicative level.

A lack of insight still exists about the way students communicate and coordinate their information processing while cooperating in a problem-solving context. When students cooperate and communicate in natural language information is exchanged, regarding the problem itself, as well as about meta-cognitive aspects such as the plausibility of the information and beliefs about the value of the information of the other.

In this chapter, we examine how 10- to 12-year-old students cooperate in two problem-solving contexts: two students working together, and a student working with a computer program. First, we discuss the advantages of a more cooperative approach for intelligent educational systems, in which the system functions as a partner and cooperates with a student who is working on a task. Implications for the design of such educational systems are addressed. Second, we discuss the results of our research on cooperative problem-solving. In this research, we collected protocols of dialogues between 10- to 12-year-old students cooperating on a problem-solving task. These protocols have been analyzed in depth with regard to the relationship between problem-solving processes and communication.

On the basis of this study's results, a prototype of a Dialogue Monitor for an intelligent cooperative system (ICS) has been implemented. This monitor is the central part of a computer-assisted educational program that can "think along" with the student and cooperates in solving a problem task. Student and system interact in a mixed-initiative dialogue that is argumentative by nature. The program has been experimentally tested with students (10- to 12-year-olds) from two elementary schools. Some of the results are described in this article. The chapter ends with a discussion of this research's implications, and our further plans for the construction of ICSs.

INTELLIGENT COOPERATIVE SYSTEMS

Three Approaches

In research into intelligent computer-assisted learning, two main approaches can be distinguished: One approach is the development of a tutoring system which acts as a teacher and guides the student and controls his or her learning path. Emphasis lies on the development of domain expert- and diagnosis-modules, and ultimately the effectiveness of such tutoring depends on instructional and curriculum expertise in specific knowledge domains. The other approach is the development of open learning environments, in which the student is able to take over control and determine his or her own learning path. These systems can take the shape of a simulation environment, a laboratory, a microworld, and so on. The effectiveness of this approach depends on the validity of a specific learning principle- that of (guided) discovery learning.

Control over the learning path lies either with the system or the student. Who has control depends neither on the ongoing problem-solving process nor the ongoing interaction process. Presently, however, a third approach seems to emerge: Research increasingly aims at a more cooperative approach (Cumming & Self, 1989). Examples are mixed-initiative systems, systems in which the student acts as a teacher to the system, advisory systems, and intelligent help-systems (Chan & Baskin, 1990; Winkels, 1992). We propose to call these systems that are meant to cooperate with the student intelligent cooperative systems (ICSs; Kanselaar, Andriessen, Barnard, & Erkens, 1990). In this approach, neither the student nor the system has complete control of the learning path. They work together as intellectual partners on a learning task (Salomon, 1988; Self, 1990).

Cooperative Learning in Education Research

Research on cooperative learning in education has a long-standing tradition. The main interest in this field was triggered by the observation that, in some circumstances, students seem to learn more from their peers than from their teachers. Besides advantages in cognitive learning, cooperative learning seems to foster social development and interpersonal (or interethnic) attitudes in the class (Johnson & Johnson, 1975; Sharan & Sharan, 1976; Slavin, 1983). The main research questions concerned the organization and effectiveness of cooperative learning in the classroom as a teaching method. Most of this research on the effectiveness of cooperative learning was directed toward the prerequisites (e.g., heterogeneous vs. homogeneous groups), the comparison with individual learning or whole-class instruction, and the final products of cooperation (Cooper & Cooper, 1984; Webb, 1982). Few researchers focused on the questions of why and how cooperative learning could facilitate learning (Doise & Mugny, 1984), or on what actually happens in the process of cooperation between students.

However, in recent educational research, cooperative learning is reemphasized (Brown & Palincsar, 1989). This emphasis follows a reformulation of learning as a social process of enculturation in recent constructivistic or situated learning views on cognition and instruction (Duffy & Jonassen, 1991; Cognition and Technology Group at Vanderbilt, 1990). Aspects of cooperation play a central role in the constructivistic approach of learning. Peer cooperation is seen, in a Vygotskian way, as an intermediate stage in the developmental process of internalization of social activities. Furthermore, notions like *cognitive apprenticeship, anchored instruction* and *scaffolding* seem to be partly based on a cooperative paradigm. The (social) learning environment should help and support the learner to construct his or her own knowledge and skills. Brown, Collins, and Duguid (1989) saw learning-both inside and outside school-advancing through collaborative social interaction and through the social construction of knowledge. They mentioned the following salient features for group learning:

1. Collective problem solving: Groups may give rise to insights and solutions that would not come about in individual situations.
2. Displaying multiple roles: Groups permit different roles needed to carry out an authentic cognitive task to be displayed by and distributed among different members in the group.
3. Confronting ineffective strategies and misconceptions: Groups may be effective in confronting and discussing faulty or non-optimal ideas of individual members.

4. Providing collaborative working skills: Group work may give the opportunity to situate experiences for future cooperative working situations.

As for the role that computers play with regard to education, the focus is on the construction of computer-based, multimedia environments: open learning environments that may give rise to multiple, authentic learning experiences (Vanderbilt Cognition and Technology Group, 1990). The cooperative aspect is mainly realized by offering computerized (intelligent) tools that can help solve the task at hand. Cooperating systems-offering tools and distributing tasks, and working together with the student-could be the next step. What should we expect of a computer-based intelligent partner in a learning context? Which criteria should a cooperative system meet?

Which Criteria Should an ICS Meet?

In natural educational settings, we can define a *cooperative learning situation* as one in which two or more students work together to fulfill an assigned task within a particular domain of learning to achieve a joint product. From this definition, the following criteria for an ICS can be inferred:

1. Complementary abilities or information: Only when the participants have abilities or information that are complementary can cooperation be fruitful. To be successfully completed, an ICS requires tasks that call for cooperation. This could also imply that an ICS does not have complete knowledge about a domain, and thus is not able to solve all the problems encountered.
2. Mixed control: In a cooperative learning situation, none of the participants is able to determine the process one sided. The participants are dependent on mutual cooperation. System and student in an ICS should have the opportunity to take control of the exchange, as well as the processing of information.
3. Mixed initiative: Both system and student have to be able to take the initiative in interaction. They must be able to take initiative in asking questions, making remarks, transferring information, suggesting solutions, and so on.
4. Common interest and common goal: In cooperation, system and student must have a common interest in solving the problem at hand. They have to reach common goals and subgoals that determine the flow of the problem-solving process.

Implications for Research and Design

Cooperation concerns a complex interaction between task strategies and communication processes. Cooperation requires that the cooperating subjects acquire a common frame of reference to negotiate and communicate about their individual viewpoints and inferences. The problem with cooperation is that the processes of representation formation and communication often take place implicitly. Natural language communication is implicit by nature-viewpoints are not always advanced, task strategies are not always open to discussion, and so forth. Although implicitness may be ineffective because it masks differences in knowledge, viewpoints, and attitudes, it also results in efficient and non redundant transfer of information. Coordination in information transfer is accomplished by multifunctional dialogue acts. With respect to an ICS, this puts a heavy burden on the interpretive power of a program. Most notably, it should deal with the functions of utterances in the situated context.

To acquire more knowledge about the coordination between communicative and problem-solving processes, it is necessary to investigate, step by step, the interaction between these processes with cooperating students. However, such an approach has been followed only scarcely within the field of cooperative learning, although the necessity of process-directed research has been expressed quite frequently (Cooper & Cooper, 1984; Webb, 1982).

Research is needed regarding how coordination between information exchange and information processing operates for students in natural learning situations. This is true for both student–student interaction and student–computer interaction. The dialogue structure analysis (DSA) project of interactive problem-solving aims to gain more insight into the relationship between processing and exchange of information in cooperative problem solving. Our central research questions are these: is the communicative process (with students in an interactive problem solving situation) determined by the problem-solving process? Conversely, how does the communicative process affect problem solving? To answer these questions, dialogues between students were analyzed within a limited, but semantically rich, domain. The goal of our research project is the development of a prototype "cooperative" computer-assisted educational program.

COOPERATION IN A PROBLEM-SOLVING ASSIGNMENT

The "Camp Puzzle"

The task that is used to study the relation between information exchange and information processing during cooperative problem solving is called the "Camp Puzzle." It is meant for students in the highest grades of elementary school (10- to 12-year-olds). The Camp Puzzle is similar to so-called "Smith, Jones, and Robinson" problems (Wickelgren, 1974). In this kind of logical problem, one has to combine different statements of information to derive some characteristics of a specified group of individuals. However, in the Camp Puzzle this task information has been split and distributed among the two cooperating partners.

By splitting this information, cooperation becomes necessary to complete the task. The cooperation partners have to exchange the relevant information, explain their reasoning, and negotiate about their inferences and task strategies.

In the instruction of the Camp Puzzle, the following situation is described: A group of six children has gone for a week-long camping holiday. Two of the children separately wrote a letter about the children in their group. The two students who work on the task are each given one of these letters. The information in each letter is insufficient to answer all the questions. For example, one letter contains the information, "The friend of Jill comes from Haarlem." In the other letter, the sentence "Ann comes from Haarlem" can be found. The students may infer that Ann is the friend of Jill, thereby ignoring the possibility that more children may come from the same city.

The students have to infer four characteristics of the six children. The collectively found solutions for the 24 subproblems can be written down in a (4×6) solution matrix. In both versions (the student–student and student–system situations), the students are allowed to work on the task for 45 minutes. The number of correctly solved subproblems can be taken as an indication of overall task performance. Although the task is perceived as difficult, the motivation and task orientation of the students (on both versions) were remarkably high.

Verbal Observation System

Protocols of the task dialogues were obtained with the aid of a semiautomatic transcription system on the basis of video recordings. This system is the verbal observation system (VOS). The VOS is a comprehensive and finely grained coding system, containing some 300 communicative and semantic coding categories. It was developed to transcribe propositional content, as well as pragmatic and communicative characteristics of utterances.

The VOS uses literal clue words in the utterances to encode the communicative function and content. The system is semiautomatic: the encoding of an utterance is being asked for by the program step by step for different variables, and the codes entered are being checked on consistency with the sets of categories defined for those variables. By the use of clue words and (limited) automatic checking, a sufficient degree of reliability in coding with this complex system could be achieved. In a reliability study with two raters coding 350 utterances, interrater-agreement percentages were found ranging from 67% to 97% for the different variables. In only 4.5% of the cases, the full encodings of an utterance were discrepant at such a degree that the communicative or semantic meaning differed in an important aspect between the raters. In the VOS, utterances are transcribed along three main characteristics: propositional content, dialogue act, and illocution.

The *propositional content* is encoded in a predicate form in which the arguments can be embedded. For example, "The friend of Jan comes from Haarlem," is represented in a form like: (city, (friend, Jan, X), Haarlem). The following types of propositions are distinguished: direct assignments, indirect references, equalities, set distributions, and axioms. The *dialogue act* represents the communicative action of an utterance. Utterances like "Does the friend of Jan come from Haarlem?", "But from Haarlem comes Jan's friend," or "No, the friend of Jan comes from Haarlem!" all have the same propositional content, but differ in dialogue act (question, counter, and denial, respectively). In the VOS 65 dialogue acts are distinguished in 19 main categories representing five communicative functions. In Fig. 10.1, the 19 main categories are given, together with their communicative function. The *illocution* represents explicitly stated illocutionary force, as described by Searle (1969). The illocutionary part of an utterance provides the listener with extra information on how to interpret the information transferred. The category system only considers explicitly stated illocution. In the Camp Puzzle, the illocution refers, in most cases, to the certainty of the information (e.g., "I am not sure that . . . ," etc.).

Protocol Analysis of Cooperative Problem Solving

The Camp Puzzle was solved by 72 pairs of students from several elementary schools. A set of 30 videotaped sessions was selected at random for analysis. The main results of protocol and statistical analyses of the cooperative task dialogues are summarized as follows.

First, on the task-content level, several logical inference procedures and task

Example:	dialogue-act:	function:
"Wait !"	command	attention signaller
"Eh, eh,..."	implicit call	
"Here it says:.."	reading aloud from letter	informative
"Ann lives in Haarlem."	statement	
"Let's write Haarlem."	proposal	elicitative
"Where does Ann live?"	open question	
"Ann from Haarlem?"	check question (yes/no)	
"Haarlem."	reply	responsive
"Ann lives in Haarlem"	repeat	
"Oh, I didn't know that"	acceptance	
"Yes, all right!"	confirmation	
"No, ...not Ann."	denial	
"Because Ann does"	reason	argument
"But Jill does not"	counter	
"Then she comes from..."	consequent argument	
"If Ann is Jill's friend."	conditional argument	
"And Ann does"	continuation	
"So Ann has to live there"	conclusion	
<writes in solution-matrix>	writing	action

FIG. 10.1. Main dialogue acts and examples.

strategies have been distinguished. The task is rather complex for the students. Unlike normal "Smith, Jones, and Robinson" puzzles, not all the solutions in the Camp Puzzle are unique. That is, more than one child can come from a certain city, engage in a certain sport, and so on. Essentially, the Camp Puzzle can be represented as a constraint-satisfaction problem, in which forward reasoning (finding positive instantiations) and elimination by constraints (finding negative instantiations) can be used to solve the problem.

Second, regarding the task dialogues, the topical structure in the dialogues coincides with the subproblem structure of the task (see Grosz, 1978). The sequence of subproblems is not rigid, and a solution path has to be found. For this purpose, topics have to be initiated, tried, and evaluated in the ongoing dialogue. Remarkably, topics are seldom explicitly proposed ("Let's search for the friend of Jan"), but are initiated implicitly by exchanging relevant information concerning a topic.

Third, most of the dialogue acts in the Camp Puzzle are informative, responsive, or argumentative (see Table 10.1). Contrary to what one would expect, the dialogues contain few open questions (e.g., "In which city does Jan live?"). The students seem to hold onto another cooperative principle (cf. Grice, 1975): "If my partner has found something interesting, he will tell me, I don't have to ask for it." Check questions (i.e., yes–no questions), are found more frequently (e.g., "Does Jan live in Haarlem?"). These questions mostly function to check information exchanged by the partner. Furthermore, the students are concerned with the plausibility or certainty of the propositions transferred or inferred by themselves or their partners (25% of the utterances have an explicit illocution part). Several plausibility levels (five in our model) can be distinguished, depending on the source of information and the depth and complexity of the inference procedure.

Fourth, on the basis of statistical sequential analyses, different topic structures and different argumentation or reasoning sequences have been identified in the protocols (Barnard, Erkens, & Sandberg, 1990). By way of illustrating the

TABLE 10.1
Difference in Percentages of Dialogue Acts Between Two Conditions

Dialogue Acts	*Student-Student (%)*	*Student-System (%)*
Statements	34	31
Supports	28	29
Questions	9	28
Denials	5	6
Arguments	20	3
Proposals	2	3
Pre-signaler	2	-

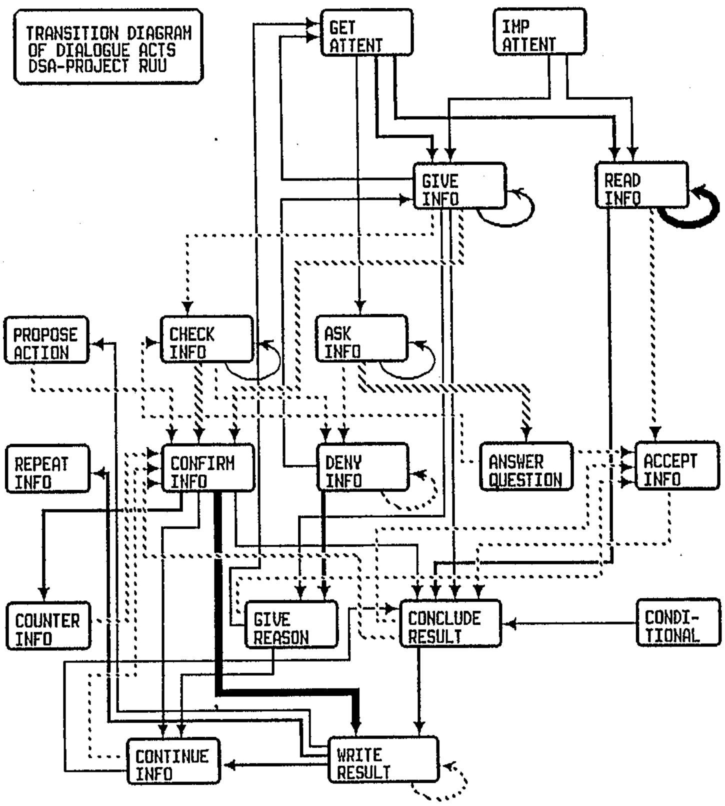

FIG. 10.2. Transition diagram of dialogue acts in student–student dialogues. The thickness of the lines indicates number of transitions, the dashed lines indicate turn taking, and the solid lines refer to auto relations.

complexity of the dialogues, a transition diagram of the student–student dialogues is represented in Fig. 10.2.

In this figure, the main transitions of dialogue acts in the students' protocols are shown. The most common pattern for the topic structure is as follows:

1. Attention signaling to the partner,
2. Or: exchange of information,
   Or: elicit information exchange,
3. Or: conclusion (of a solution),
   Or: support of information by the partner,
   Or: check of information by the partner, followed by:
      Or: confirmation,

Or: discussion with counterarguments, denials,
explanation, and reasons,

4. Or: responding to the question for information, followed by:
   Or: support of the reply,
   Or: checking of the reply,
5. Or: writing a solution in the matrix and continuing,
   Or: continuing with new information transfer.

The prototype of the dialogue monitor has been developed based on these and other findings in the dialogue protocols.

COOPERATION WITH THE COMPUTER

Model of Cooperative Problem Solving and Dialogue Processing

The kind of task being discussed here, in which information exchange is central, contains a complex relationship between the problem-solving and dialogue processes. The DSA model of cooperative problem solving and information exchange is based on our analyses of dialogue protocols and similar approaches in the literature (in particular, Carberry, 1985; Fortescue, 1980; Grosz, 1978; Reichman, 1985). The model contains a number of cognitive information-processing subsystems, and specifies the predicted relations between these subsystems as part of the individual student in his or her interaction with the outside world, including the task partner (see also Barnard & Erkens, 1989).

The DSA model is reflected in the modular architecture of the prototype Dialogue Monitor for an ICS. In principle, the Dialogue Monitor is meant to be used for cooperative problem solving in different domains of declarative knowledge and logic. A first version of the Dialogue Monitor has been implemented for cooperation with the Camp Puzzle. Following a description of the Dialogue Monitors architecture, we discuss some of the first results of experimenting with the program.

Architecture of the Dialogue Monitor Program

The Dialogue Monitor computer program contains five different modules, each with a specified function. The program's architecture is presented in Fig. 10.3, and represents our model of problem-solving and information exchange for a single student in a cooperative-task situation.The program is programmed in Prolog and runs on 80286 or higher IBM/MS-DOS machines.

In this model, the *external world* represents the external sources of information with which the monitor communicates: the cooperation partner and the task

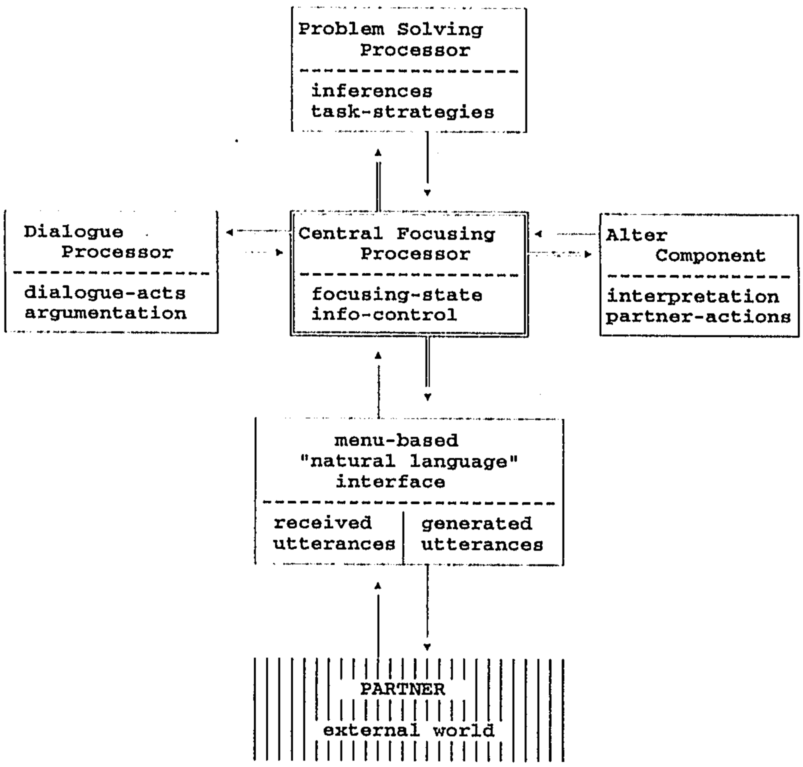

FIG. 10.3. Architecture of the dialogue monitor.

material (letter and solution matrix). The *interface* is the channel through which communication with the outside world (i.e., the cooperation partner) takes place. Internally, four components are assumed to process incoming and outgoing information. The arrows indicate the internal-information exchange between the various components. The double arrows from the central-focusing processor indicate a control function.

The *problem-solving processor* contains knowledge of problem-solving procedures (i.e., rules indicating which [sub]problems are to be tackled). About 40 different logical-inference procedures and subprocedures are implemented in this module. Two faulty inferences are also implemented, and these can be optionally addressed. All intermediate steps are being saved as frames in a limited working memory, with an estimation of the certainty level of the inference itself. These frames can be used later for explanation or question answering. In this module, task strategies (which subproblem next?) and information evaluation (what kind of information is interesting?) are also performed.

The *dialogue processor* contains knowledge about dialogue acts, (i.e., about the forms of the utterances by which information is exchanged and reactions from the partner can be elicited). The dialogue processor also can interpret the communicative function of the dialogue acts coming from the partner. Furthermore, the dialogue processor contains argumentative and explanatory scenarios.

The *alter component* contains some inference rules from which a picture of the partner's current cognitive and communicative activities might be derived. For example, a partner's silence could lead to the conclusion that he or she is busy looking for information in his or her own letter.

These three components operate on information stored in frame-based working memories, which contain the currently relevant information about problem solving, dialogue process and partner.

The *central-focusing processor* is the central part of this model: It has a monitoring function for the internal coordination and working of the program as a whole. This component determines the flow of information between the various components, as well as from and toward the outside world. The focusing processor combines the results of the various components. The general task of the central-focusing processor is to interpret and check incoming utterances of the partner, and to generate utterances in a reacting or initiating way. An important task in this context is determining the focus and topic of the current dialogue context.

The importance of checking the information put forward by the partner is reflected in our simulation model by a checking procedure that operates on every incoming utterance of the partner. By this checking procedure, the information transferred is compared with the knowledge base, and the plausibility of this new information (i.e., the credibility) is estimated before the information is accepted and used for further inferencing. Besides checking questions, this procedure accounts for most of the confirmations, acceptances, repeats, denials, and counters observed in the protocols.

For the actual interaction with the system a menu-based "natural language" interface has been constructed (see Miller, 1988; Tennant, Ross, Saenz, Thompson, & Miller, 1983). By means of interconnected menus, the student can select different constituents of the utterance he or she wants to make. Connective, type of sentence, subject, predicate (i.e., verb), object, and illocution can be picked separately and repeatedly. After each selection, the utterance made so far is being updated in a grammatically correct, "natural language" form and shown in a window. In this way, the interface is flexible and easy to use. With the interface, many different sentences can be made (about 3.2 million). In working with the interface, the student makes, in reality, a proposition in the internally used VOS representation of the program. A separate module translates these propositions, as well as the propositions generated by the monitor, in "natural" (Dutch) sentences. The advantages of this kind of interface should be obvious: No ambivalent semantic parsing and no typing skills are required. In Fig. 10.4, an untrans-

```
SOORT ZIN PERSOON/GROEP  KENMERK WEL/NIET ZEKERHEID KLAAR
                        in  GROEP.. zitten
                        uit STAD..  komen
                        aan SPORT.. doen
                        is BEVRIEND met
                        een (J/M).. zijn Piet
                           NAAST..slapen Ans
                        zelfde GROEP als Jan
                        zelfde STAD  als Joke
                        zelfde SPORT als Kees
                        ook BEVRIEND met Els
                        is ook (J/M)...  [en-GROEP]namen
                                         [of-GROEP]namen
                                         elkaar

Maar Piet en Jan slapen naast.
```

FIG. 10.4. Screen dump of the menu-based language interface.

lated screen display of the interface is shown. The sentence, "But Piet and Jan sleep next to (each other)" is being made.

Procedure

After a short instruction by the experimenter, the student receives a letter (in an envelope) and is asked to type his or her first name. The rest of the interaction with the system takes place by means of menus. The program asks which letter the student received so that he or she can take the other one. Subsequently, use of the interface is trained by an instruction module. After this instruction, the cooperation session starts. The screen is divided into three parts, from top to bottom: solution matrix, dialogue window, and sentence window of the interface. In the dialogue window, the current dialogue scrolls up. The menu-based, "natural language" interface can be activated at any time by any key. On activation of the interface, interaction and initialization of dialogue acts by the program stop until the student has finished his or her sentence. Task dialogues are logged and stored by the program.

SOME RESULTS ON STUDENT-SYSTEM INTERACTION

The program has been experimentally tested on 40 students in the highest grade of two elementary schools (10- to 12-year-olds). The students cooperated with the program under four different conditions, representing optional levels of task content and communicative expertise. On the task-content level, low expertise was simulated by also allowing the system to make faulty (bi-implicational) inferences. Different levels of communicative expertise concern the tendency of the monitor to check or accept the information exchanged by the partner. We discuss the main preliminary results and our first experiences with the program in this section. We concentrate on the comparison between student–student and student–system interactions. A full report of the experiment's results is given in Erkens (in press).

First, and most important of all, the program was able to maintain a cooperative dialogue with a student for 45 minutes (adjusted session period) without too many bugs or inconsistencies. Figure 10.5 represents a dialogue fragment of the monitor's interaction with a 12-year-old student. The fragment is literally translated from Dutch. Utterances made by the student are prefixed with alter. The sign "[]" at the end of a line indicates a pause of more than 2 seconds.

```
 alter   =  12 year-old male student        [] = pause >
 monitor =  Dialogue Monitor                      2.0 sec.
-----------------------------------------------------------------
 1 alter   : "Chris does gymnastics;
              that is in my letter."                           []
 2 monitor: "Does Chris do gymnastics ?"                        []
 3 alter   : "Yes."                                             []
 4 monitor: "So Chris does gymnastics."
 5 alter   : "Yes."                                             []
 6 monitor: "Oh..."                                             []
 7 monitor: "We can write that : Chris on gymnastics."
 8 monitor: "Well, I have written it."
               [ie. filled in the solution-matrix]
 9 monitor: "All right !"                                       []
10 alter   : "Let's search : the partner of Jan."               []
11 monitor: "Yes..."
12 monitor: "With whom does Jan work together ?"
               [i.e. cooperate with]                            []
13 alter   : "And the partner of Jan is a girl;
              that is in my letter."                            []
14 monitor: "Oh..., the partner of Jan is a girl                []
15 alter   : "Yes; that is correct."                            []
16 monitor: "Oh..."                                             []
17 alter   : "From which city does Jan come from ?".            []
18 monitor: "The city of Jan I don't know;
              it is not in my letter."                          []
19 alter   : "Jan does the same sports as Jill;
              I am sure of that."                               []
20 monitor: "So, Jan does the same sports as Jill."             []
21 monitor: "Jan does the same sports as Jill;
              you said that."
22 monitor: "Jill does swimming; that is in my letter."
23 monitor: "Then Jan does swimming."
24 monitor: "We can write that: Jan on swimming."
25 monitor: "It is written !"
               [ie. filled in the solution-matrix]
26 monitor: "What else do you know ?"                           []
27 alter   : "Wait a minute...
```

FIG. 10.5. Fragment of a dialogue protocol of a student and the dialogue monitor.

As can be seen in the fragment, the dialogue has a rather "natural" flavor with a lot of implicit proposals, ellipsis ("Yes," "Oh"), nonspecific replies (13), plausibility checking (2), and unexpected topic shifts (17, 19). In fact, most dialogues, look, at first sight, rather similar to student-student dialogues which is promising for the aims of our research. Although in this fragment the student takes most of the initiative, this is not a characteristic feature of all dialogue protocols (in fact, this student was too quick for the monitor to be able to initiate).

The analyses also revealed some problems in using the program, which may complicate the comparison of the student–student version with the student–computer version of the task. One important drawback is the slowness of written interaction. The student–system dialogue needed much more time than the verbal speech dialogues between students. In the student–student version, the Camp Puzzle was solved in about 25 minutes. In addition, the average score of correctly solved subproblems in a student–student version was 20.5, whereas in the student–system version the average score was only 13.2. The lower score is not caused by an abundance of mistakes and faulty solutions, but simply by the fact that the students are not able to complete the task in the 45 minutes.

Another problem was that the students do not seem to make full use of the interface. They tend to stick to the same sort of sentences; they do not vary much, especially in the first half of the session. The time allowed for instruction and exercises with the menu-based interface is probably too short. Most students only seem at ease with the interface in the last 10–15 minutes of the session. Only then do they start to experiment with the sentences they construct, use various connectives and illocutions, and seem more comfortable with the program as a whole. This problem could be solved by a separate instruction session to get familiar with the program and its interface.

There may be another factor involved, which is perhaps more fundamental and disturbing to our research. Quite a few students seem to be impressed by the program—they seem to comply with the actions of the monitor and take little initiative themselves. In short, they do not seem to cooperate with an equal partner, but with an authority (see Table 10.1). Although the program can make mistakes as well, most mistakes made in the low-expertise condition are accepted or believed by the students. As can be seen in Table 10.1, the percentage of questions in the student–system condition are rather high compared with student–student dialogues on the same task. On the whole, the number of denials, arguments, and proposals in the student–system condition is very low. In contrast to this, there are more argumentative utterances in the student–student interaction.

Another complicating factor may be an artifact of the program. The constraint put on the dialogue for the students is that only one utterance at the time is accepted. Accordingly, the length of each student's turn in the dialogue is set to one utterance. Although the student is able to relate an argument to another argument in his or her next turn, he or she is not allowed to build an "argumentation" in a sequence of directly connected arguments. We did not observe students to have problems or frustrations concerning this constraint. Still, the constraint causes asymmetry in the dialogues because the monitor bypasses this constraint. The constraint was implemented in the program out of fear of combinatorial explosion when interpreting multiple utterances.

FINAL REMARKS

Presently we have not fully analyzed he student–system dialogues collected with the Dialogue Monitor. A more elaborate comparison with the student–student dialogue protocols also need to be made. These results will further improve the program. The main improvements were referred to in the results section. They encompass a longer period of instruction and more exercises with the menu-based, natural language interface. By these means, the students will be better prepared for cooperation with the program. Furthermore, we plan to solve the direct interpretation of student utterances on the part of the Dialogue Monitor, also giving the student the opportunity to make multiple utterances in one turn.

Further developments could be implementing the Dialogue Monitor on another domain of learning (biological taxonomy), and experimenting with speech synthesis to get spoken utterances by the Dialogue Monitor. To be sure, the Camp Puzzle knowledge domain of the Dialogue Monitor is restricted and, furthermore, rather artificial in the school curriculum. In the construction of the Dialogue Monitor, we have tried to separate the domain-dependent knowledge and inference procedures to enable the use of the program in other domains. We already developed a similar cooperation task on biological taxonomies. An implementation of the program on this domain would test our expectation of the potential generalizability of dialogue processes in cooperative problem solving.

Although the students did not argue very much with the program, the argumentation of the program is not optimal either. As yet, we are not satisfied with the rigid, rather standardized way the program acts in situations of contra-argumentation. This is a difficult problem to solve because it has to do with belief revision. In contra-argumentation, in the case of disagreement on a line of reasoning, one of the cooperation partners should be convinced in the end (Van Eemeren & Grootendorst, 1984). The problem is how to specify the conditions under which the program will have to lose its faith in its own line of reasoning. It is hard to specify when and how the program will have to revise its knowledge—not only some factual knowledge, but a complete inference procedure. More research on natural argumentation and conviction will be needed to simulate these processes in a cooperative system. In general, we think this may be one of the main problems in future development of ISCs. Dynamic adaptation of reasoning procedures is of the utmost importance because it has to do with the core of current views on learning and interactivity: the negotiability of knowledge (Winograd & Flores, 1986). In conclusion, the program and session procedure will have to be improved in several respects. However, the nature of the task dialogues between students and systems already obtained gives rise to optimism about the possibility to construct ICSs.

ACKNOWLEDGMENT

This research was funded by the Institute for Educational Research (SVO-project 203.7003).

REFERENCES

Barnard, Y. F., & Erkens, G. (1990). Simulation and analysis of problem solving and dialogue processing within cooperative learning. In H. Mandl, E. De Corte, N. Bennett, & H. F. Friedrich (Eds.), *Learning and instruction. European research in an international context* (pp. 181–196). Oxford: England: Pergamon Press.

Barnard, Y. F., Erkens, G., & Sandberg, J. A. C. (1990). Interaction in intelligent tutoring systems. *Journal for Structural Learning, 10,* 197–213.

Brown, A. L., & Palincsar, A. S. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. B. Resnick (Ed.), *Knowing, learning and instruction* (pp. 393–451). Hillsdale, NJ: Lawrence Erlbaum Associates.

Brown, J., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning, *Educational Researcher, 18,* 32–41.

Carberry, M. S. (1985). *Pragmatic modelling in information system interfaces.* Unpublished doctoral dissertation, University of Delaware, Newark, DE.

Chan, T. W., & Baskin, A. B. (1990). Learning compagnion sytems. In C. Frasson & G. Gaulthier (Eds.), *Intelligent tutoring systems: At the crossroad of artificial intelligence and education* (pp. 6–34). Norwood, NJ: Ablex.

Cognition and Technology Group at Vanderbilt (1990). Anchored instruction and its relationship to situated cognition. *Educational Researcher, 19*(6), 2–10.

Cooper, C. R., & Cooper, Jr. R. G. (1984). Skill in peer learning discourse: What develops? In S. A. Kuczaj II (Ed.), *Discourse development. Progress in cognitive development research* (pp. 77–97). New York: Springer.

Cumming, G., & Self, J. (1989). Collaborative intelligent educational systems. In D. Bierman, J. Breuker, & J. Sandberg (Eds.), *Artificial intelligence and education: Proceedings of the 4th International Conference on AI and Education* (pp. 24–26) Amsterdam: Netherlands, IOS.

Derry, S. J. (1992). Metacognitive models of learning and instructional systems design. In M. Jones & P. H. Winne (Eds.), *Adaptive learning environments: Foundations and frontiers* (pp. 257–287). Berlin/Heidelberg: Springer-Verlag.

Doise, W., & Mugny, G. (1984). *The social development of the intellect.* New York: Pergamon.

Duffy, T. M., & Jonassen, D. H. (1991, May). Constructivism: New implications for instructional technology? *Educational Technology* (pp. 7–12).

Van Eemeren, F. H. & Grootendorst, R. (1984). *Speech acts in argumentative discussions: A theoretical model for the analysis of discussions directed towards solving conflicts of opinion.* Dordrecht, Holland: Foris.

Fortescue, M. D. (1980). *A discourse production model for "Twenty Questions."* Amsterdam: John Benjamins.

Grice, H. P. (1975). Logic and conversation. In P. Cole & J. Morgan (Eds.), *Syntax and semantics* (pp. 41–58). New York: Academic Press.

Grosz, B. J. (1978). Discourse knowledge. In D. E. Walker (Ed.), Understanding spoken language (pp. 229–344). New York: Elsevier-North-Holland.

Johnson, D. W., & Johnson, R. T. (1975). *Learning together and alone*. Englewood Cliffs, NJ: Prentice-Hall.

Kanselaar, G., Andriessen, J. E. B., Barnard, Y. F., & Erkens, G. (1990). Some issues on interaction in intelligent cooperative systems. In J. Pieters (Ed.), *Intelligent tutorial systems and instruction* (pp. 45–65). Enschede, The Netherlands: Twente University.

Miller, J. M. (1988). The role of human-computer interaction in intelligent tutoring systems. In M. C. Polson & J. J. Richardson (Eds.), Foundations of intelligent tutoring systems (pp. 143–184). Hillsdale, NJ: Lawrence Erlbaum Associates.

Reichman, R. (1985). *Getting computers to talk like you and me; discourse context, focus and semantics (an ATN model)*. Cambridge, MA: MIT Press.

Salomon, G. (1988). *Technology of the mind: Partner and nourisher.* Paper presented at the Educational Research Conference, Leuven, Belgium.

Searle, J. R. (1969). *Speech acts, an essay in the philosophy of language*. London: Cambridge University Press.

Self, J. A. (1990). Bypassing the intractable problem of student modeling. In C. Frasson & G. Gaulthier (Eds.), *Intelligent tutoring systems: At the crossroad of artificial intelligence and education* (pp. 107–124). Norwood, NJ: Ablex.

Sharan, S., & Sharan, Y. (1976). *Small-group teaching*. Englewood Cliffs, NJ: Educational Technology Publications.

Slavin, R. E. (1983). *Cooperative learning*. New York: Longman.

Tennant, H. R., Ross, K. M., Saenz, R. M., Thompson, C. W., & Miller, J. R. (1983). Menu-based natural language understanding. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics* (pp. 151–158). New York: Association of Computing Machinery.

Webb, N. M. (1982). Student interaction and learning in small groups. *Review of Educational Research, 52*(3), 421–445.

Wickelgren, W. A. (1974). *How to solve problems: Elements of a theory of problems and problem solving*. San Francisco: Freeman.

Winkels, R. (1992). *Explorations in intelligent tutoring and help*. Amsterdam: IOS Press.

Winograd, T., & Flores, F. (1986). *Understanding computers and cognition: A new foundation for design*. Norwood, NJ: Ablex.