In Intelligent Tutoring Systems, Proceedings of the 5th International Conference (ITS 2000), G. Gauthier, C. Frasson, & K. VanLehn (Eds.), pp. 325-333. Berlin: Springer-Verlag.

A Coached Collaborative Learning Environment for Entity-Relationship Modeling

María de los Angeles Constantino-González [1] and Daniel D. Suthers [2]

[1] Center for Artificial Intelligence, Monterrey Institute of Technology (ITESM), E. Garza Sada 2501 Sur, Monterrey, N.L. 64849, Mexico
aconstan@cia.mty.itesm.mx

[2] Information and Computer Sciences, University of Hawai'i, 1680 East West Road, POST 303A, Honolulu, HI 96822, USA
suthers@hawaii.edu

Abstract. We discuss the design of an agent for coaching collaborative learning in a distance learning context. The learning domain is entity-relationship modeling, a domain in which collaborative problem solving is regularly practiced, and for which there exist formally interpretable representations of problem solutions known as entity-relationship diagrams. The design of the coach was based on socio-cognitive conflict theory, which states that collaborative learning is effective to the extent that learners identify and discuss conflicts in their beliefs. Students begin by constructing individual entity-relationship diagrams expressing their solution to a database modeling problem, and then work in small groups to agree upon a group solution. The coaching agent leverages learning opportunities by encouraging students to share and discuss solution components that conflict with components of the group solution. Our work shows one way to utilize domain specific knowledge in order to facilitate collaboration.

1 Introduction

Distance Learning has become a very popular educational paradigm due to recent advances in computing and telecommunications, and shifts in the demographics of student populations to include more working or geographically isolated individuals. Unfortunately, distance learning has focused primarily on individual learning and has given little emphasis to the development of collaborative skills. Although students sometimes work in a group, there is little evaluation of the collaboration process and the students' collaborative skills. Yet, "with social and intellectual isolation, students may fail to develop and refine those cognitive an interpersonal skills increasingly necessary for business and professional careers" [1]. The development of these skills should be promoted in distance learning environments.

Studies of collaborative learning in the classroom show that, properly designed, collaborative learning improves students' achievement, critical thinking and cooperative behavior [9, 13]. However, just as simple interaction in the classroom does not

equal collaboration [7], it is not sufficient to provide distance learners with only a channel of communication. Individuals may not consider others' opinions, and may give only a final solution without explaining how that solution was obtained [23]. Coaching in a computer-mediated collaborative environment may help distance students to develop collaborative skills.

Several knowledge-based systems have been designed to support the development of group skills. Some of them were developed for general discussion scenarios, typically comparing use of sentence openers to models of dialogue [14, 15, 18, 20]. Some systems track both dialogue and domain actions [4], or provide individual domain specific tutoring in a collaborative virtual environment [10]. Finally, other systems have been developed to support collaboration by pairing students who need help with those who can supply that help [2, 11]. The difference between our software coach and the previous work discussed above is that our coach utilizes domain specific knowledge in order to facilitate collaboration.

Our coach supports computer-mediated collaborative learning of database design. It functions within a “groupware” environment in which students construct solutions to database modeling problems using the Chen diagrammatic notation [8] for Entity-Relationship (ER) Modeling. ER modeling is one of the most commonly used data modeling formalisms for conceptual database design, a collaborative task in which analysts and database users participate to produce a conceptual schema of their global view of data [5, 10]. The performance of the final database system depends highly on the correct design of the conceptual schema, yet data modeling is a difficult task for novices [6, 19].

The coach identifies semantically important differences between students' ER diagrams and, based on neo-Piagetian theory concerning the role of conflict and its resolution in learning, encourages students to address these differences in ways expected to lead to learning. According to Socio-Cognitive Conflict Theory, students learn from controversies when they discuss their different viewpoints, pose alternatives, and request and give explanations [24]. The value of the disagreement does not depend as much on the correctness of the opposing position as on the attention, thought processes and learning activities it induces [12]. Considering the utility of cognitive conflicts, it is important that students confront and discuss their differences. Unfortunately, this kind of interaction may not arise spontaneously [17, 24]. Collaborative distance learning software should provide an environment where students are encouraged to make their ideas explicit to others, be aware of and review their teammates participation, give and request explanations, and address conflicts. Our software is intended to facilitate these kinds of interactions.

2 COLER

COLER is a World Wide Web (WWW)-based computer-mediated **Co**llaborative **L**earning environment for **E**ntity **R**elationship modeling. The learning objectives are to improve students’ performance in database design using the Entity-Relationship modeling formalism, and to help students develop collaborative and critical thinking

In Intelligent Tutoring Systems, Proceedings of the 5th International Conference (ITS 2000), G. Gauthier, C. Frasson, & K. VanLehn (Eds.), pp. 325-333. Berlin: Springer-Verlag.

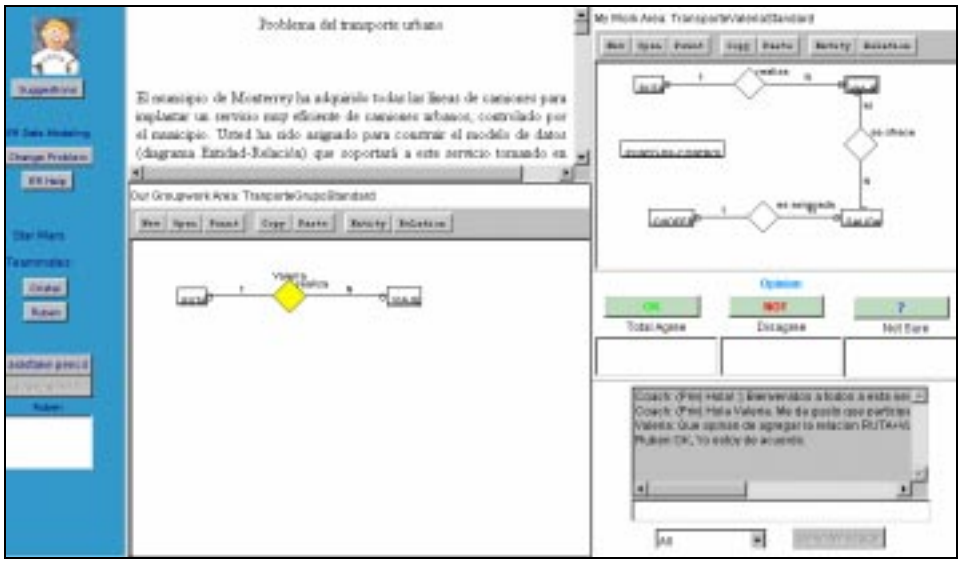

Figure 1. COLER Group Session Interface

skills. The design objectives are to promote learning interactions via an intelligent coach, to enable interaction between students from different places via a networked environment, and to not be restricted to a specific platform.

2.1 COLER's Interface

COLER provides four different modes of operation according to the type of user (student/professor) and the selected type of session (individual/group). COLER's student/group interface is shown in Figure 1. The *problem description window* (upper center) presents an entity-relationship modeling problem. Students construct their individual solutions in the *private workspace* (upper right) using the Chen notation. They use the *shared workspace* (lower center) to collaboratively construct ER diagrams while communicating largely via the *chat window* (lower right). They can use a **HELP** *button* (upper left) to get information about Entity-Relationship Modeling. A *team panel* (middle left) shows which teammates are already connected. Only one student, the one who has the pencil, can update the shared workspace at a given time. The *floor control panel* (bottom left) provides two buttons to control this workspace: **ASK/TAKE PENCIL** and **LEAVE PENCIL**. Additionally, this panel shows the name of the student who has the control of this area and the students waiting for a turn. An *opinion panel* (middle right) shows teammates' opinions on a current issue. This area contains three buttons: **OK**: Total agreement, **NOT**: Total or Partial Disagreement, and **?**: Not sure, Uncertainty. When a button is selected, students have the option of annotating their selection with a justification. Opinion button selections are displayed in both the chat area and the opinion panel. We insert opinion selections (along with any optional justifications) into the chat in order to correlate these opinion-expressing

actions with the chronology of the chat discourse. We display selections in the opinion panel to provide students with a persistent summary of their teammates' current opinions. A *personal coach* (upper left) gives advice in the chat area based on the group dynamics: students participation and group diagram construction. Although several suggestions may be computed at a certain time, only one is shown in the chat area. The others may be given on demand by pressing the **SUGGESTIONS** button, which is disabled if the coach does not have any advice to offer.

COLER is designed for sessions in which students first solve problems individually, and then join into small groups to develop group solutions. The initial problem solving helps ensure individual participation, and provides the raw materials for the negotiation of differences. The private workspace also enables students to try solutions they are uncertain of without feeling they are being watched. When all of the students have indicated readiness to work in the group, the shared workspace is activated, and they can begin to place components of their solutions in the workspace. This may be done either with **COPY/PASTE** from private workspaces, or by making new structures in the shared workspace. After each change to the workspace, the changed object is highlighted in yellow; then, students are required to express their opinions using the **OK/NOT/?** buttons before making subsequent use of the shared workspace.

2.2 Implementation Architecture

COLER's implementation is based on an architecture for intelligent collaborative learning systems originally used for the implementation of the Belvedere software for collaborative critical inquiry [22]. The COLER implementation architecture includes Java 1.1 applets that are in charge of the different functions of the system, such as chat, floor control, voting, private and shared ER modeling. It uses Belvedere's diagrammatic classes and components for networking. COLER's applets as well as the personal coach communicate with an mSQL database server via a JDBC Object Request Broker. This broker also informs the Connection Manager of user changes. The Connection Manager is a process on the server that keeps track of the clients using any diagram. It informs other clients via their Listener sockets of the changes to their diagram for "what you see is what I see" updating.

2.3 Coach Modules

The main activity of the coach is to monitor participation and to identify and evaluate differences between diagrams to encourage students to discuss them. Each student's client contains a private coach, which monitors the private workspace of its student - the "currently monitored student" or CMS. Each coach also monitors the shared workspace, and records students' selections of opinion buttons and their participation in the shared workspace and in chat discussions. (No natural language interpretation is attempted.) The Coach utilizes four modules, each contributing distinct expertise, and each implemented as a Java thread.

In Intelligent Tutoring Systems, Proceedings of the 5th International Conference (ITS 2000), G. Gauthier, C. Frasson, & K. VanLehn (Eds.), pp. 325-333. Berlin: Springer-Verlag.

Differences Recognizer. Opportunities for students to collaborate are detected by finding semantically significant differences between individual and group ER diagrams. The Differences Recognizer can either find differences specifically related to the currently added object, or find all "extra work" that the student can contribute to the group. The differences were identified through a review of database literature and with the guidance of a database expert. A weight is assigned to each difference depending on its impact on solution quality. This weight is considered by the coach to decide when to give advice. Top-weighted differences include Missing entity, Extra entity, Missing relationship, Extra relationship, Missing key attribute, Difference in cardinality, and Difference in optionality.

Diagram Analyzer. This module detects ER diagram anomalies. Currently it is only syntax-based. Later we will check for semantic anomalies.

Participation Monitor. The Participation Monitor attends to the activity in the group diagram. If nobody has worked in the group diagram for a period of time, it reports this event. It also monitors whether each student is participating too much or too little. The monitor tracks each student's number of contributions, incrementing the value each time a student adds something. To evaluate participation, a standard deviation (s.d.) is computed. If the s.d. exceeds a threshold of participation (e.g. 1.5) the monitor assumes there is a problem in participation and individual students are checked. If the difference between a given student's activity and the mean exceeds the s.d., then it is assumed that the student is part of the problem, as follows: Student Contributions - Mean > s.d.: Student has participated too much; Mean - Student Contributions > s.d.: Student has not participated enough.

Personal Coach. The Differences Recognizer, Diagram Analyzer and Participation Monitor communicate their results to the Personal Coach, which generates potentially applicable advice and selects the advice to give, if any. This process is described in detail below.

2.4 Generating Advice

Advice types were defined based on the collaborative learning literature and several observatory studies summarized in a subsequent section. The types (and abbreviations used in Figure 2) are: *Group participation* (encouraging participation and/or allowing others to participate): Contributing to the group solution (GC), Contributing a specific object to the group solution (SC), Giving teammates a chance to participate (LP, LM), Listening to others (LO), Inviting others to participate (IP), Continue working on task (not shown in Figure 2), General Participation (GP). *Group discussion* (suggestions for chat): Ask for or Give explanations or justifications (AE, GE), Analyze alternative solutions (AA), Express disagreement (ED), Express uncertainty (EU). *Feedback* (suggestions for use of opinion buttons): Give feedback by voting (GF), Ask for feedback (AF). *Reflection*: Reflect about a specific issue (RA). *Checking Own Discrepancies:* Review one's contribution (CD). *ER Modeling* (diagram syntax): Connect a disconnected entity, define an entity's key, add a relationship's name, review whether the diagram is complete (not shown in Figure 2).

In Intelligent Tutoring Systems, Proceedings of the 5th International Conference (ITS 2000), G. Gauthier, C. Frasson, & K. VanLehn (Eds.), pp. 325-333. Berlin: Springer-Verlag.

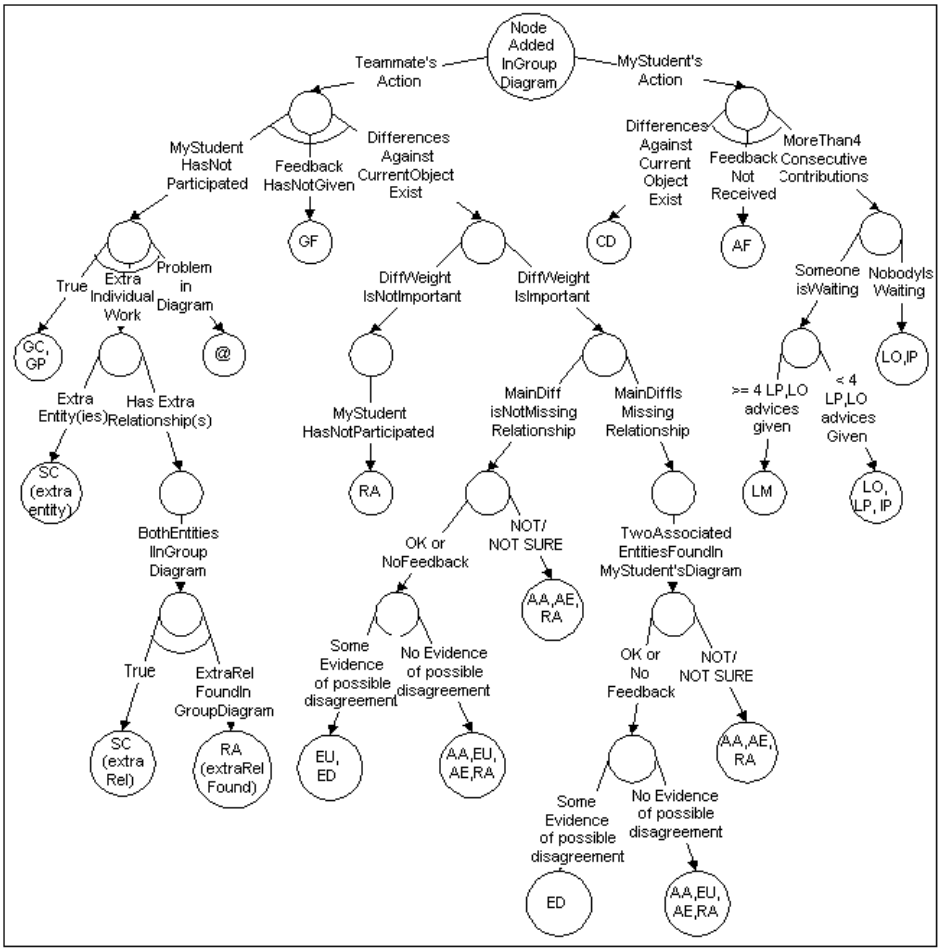

Figure 2. A portion of the coach's decision tree

The coach might generate several suggestions for any given event. For example, a decision tree describing the coach's reasoning for the "Add Object" event is shown in Figure 2. This is an and/or decision tree: as indicated by the "and" arcs, several leaves may be reached at once. Also, each leaf may propose more than one advice type (AA, EU, etc.). The tree has two main "or" branches, depending on who performed the action.

If a teammate added the object (left hand branch of Figure 2), processes to evaluate (1) participation, (2) feedback, and (3) discussion are executed (branches connected by an "and" arc): (1) If the Participation Monitor indicates that the CMS has not participated, the coach generates advice shown in the leftmost subtree: General Contribution, General Participation, Reflection About an issue. Specific Contribution suggestions are computed according to extra individual work done by CMS. ER Modeling suggestions correspond to syntax-based problems found in the group dia-

gram, according to a subtree, not shown, attached at the @ in Figure 2. (2) Feedback evaluation simply checks whether the student has selected one of the **OK**/**NOT**/**?** buttons. (3) The discussion evaluation process takes into account the number of differences found, the type of differences and the student's participation. Discussion suggestions are generated by the evaluation of the differences found, as shown in the large subtree in the center of the figure.

If the CMS added the object (right hand branch of Figure 2), processes to evaluate (4) discrepancy checking, (5) received feedback and (6) participation are executed: (4) A suggestion to Check Discrepancies is generated based on an evaluation of the differences found. (5) If opinions from others (OK, NOT, ?) have not been received, then Ask for Feedback is generated. (6) If the Participation Monitor indicates that the CMS has done many consecutive contributions, it generates advice from the following types: Listen to Others, Let Participate, Invite others to Participate.

2.5 Selecting Advice

Sets of advice can potentially be generated by many of the leaves of either the left or right subtree of Figure 2, as well as by trees for other events. If the combined set includes several types of advice for the same category, those that have been previously used are eliminated, implementing a preference to avoid giving the same kind of advice twice. If there is no new type of advice, all the types of advice are considered. Then, for each selected type of advice an advice pattern is randomly chosen and the corresponding text is generated by binding variables from the current situation (e.g. student's name, object's type, object's name). Finally, a preferences-based sort algorithm [21] is run if needed to choose between multiple advice instances. Examples of preferences are New Advice (don't repeat variable bindings), Many Instances (prefer advice of a type that applies more than once), and Category Preferences (e.g. Discussion, Participation). The order of the preferences may change during the collaborative session depending on the group's performance. If the group seems to need more participation advice, this category of advice is promoted. Otherwise, discussion is encouraged.

3. Empirical Studies

The design of both COLER's interface and coaching algorithms has been based on numerous empirical studies, to be described in detail in a future publication. Initially we observed small groups of students solving ER problems using pencil and paper, in both laboratory and classroom settings, in some cases with a database expert monitoring and coaching their work. These studies helped us understand the range of group dynamics and of responses to coach interventions. Once the COLER interface became available, we conducted "Wizard of Oz" studies in which the expert could see students' individual diagrams and send chat messages to both individuals and the group. In these sessions we observed the types of advice given by the expert and

students' responsiveness to this advice. Students were usually cooperative, but we also identified the need to enable the coach to take away the pencil from a student who is not letting the others participate. Informed by these paper-based and Wizard of Oz studies as well as by our literature review, COLER's coach was then implemented by the first author.

The automated coach is currently undergoing testing. Five further sessions have been conducted in order to evaluate usability and the coach algorithms. Teams of three students participated in each of these sessions. Some of these students were taking a database course at ITESM; others were graduate students who had already taken this course. Students located in different rooms used COLER to solve a database modeling problem first individually and later in a group. At the end of the two-hour study they answered a survey regarding the performance of the coach.

Two of these sessions were conducted with the human expert present, who was able to comment on the performance of the coach. These sessions helped us to detect some problems in COLER's user interface and coach algorithms. The software coach was pressing students too much to continue working when some time had passed and they had not done any action in the group diagram. Some of these problems were fixed by adjusting the values of parameters.

The other three sessions were conducted with the human coach not present. Instead, he is being shown the information available to the automated coach, and asked to rank the advice types according to their reasonableness for each situation. These judgements will be compared to the advice actually generated and selected by COLER in order to evaluate the advice generation and selection algorithms. This evaluation is underway at this writing.

From the surveys we learned that students think that the presence of a coach during the session is important to motivate them to discuss and help others; that the coach didn't interrupt them too much (after the adjustments); and that some of the advice was useful while some was irrelevant or redundant. They asked for some group advice instead of just personal suggestions and to include some domain-advice. We are conducting further studies to observe how students will use COLER's advice.

4. Conclusions and Future Work

The current version of the coach only has access to the student's private workspace and the shared workspace. If a portion of the student's solution differs in a nontrivial way from the group's solution and the student has not disagreed with the group's solution this coach highlights the difference and suggests that the student discuss it with the others. Thus, this version of the coach helps *avoid "missed opportunities"* for collaborative learning [3]. A planned future version of the coach will differ primarily in its ability to inspect all students' private workspaces as well as the shared workspace. This coach will notice when two students have conflicting solutions, neither of which have been shared with the group. It will be able to suggest that one of the students share his/her solution with the others. If the student does so, the conflict situation that would result can be addressed by the first version of the coach.

Thus, this version of the coach will help to *create* opportunities for collaborative learning. Each one of these versions can be improved by adding knowledge in the form of a pre-stored "expert" or correct solution to the problem being solved. This additional knowledge may improve the coach's selection of which collaboration advice to give and to whom, as well as to comment on correctness of solutions [16].

This work is part of a research agenda that seeks to characterize the knowledge needed to facilitate collaborative learning processes. We focus on how domain specific reasoning can guide the coaching of collaborative interactions. (Other work has either coached domain problem solving or provided generic support for collaborative dialogues.) Specifically, we are investigating how much leverage can be gained by a basic ability to detect semantically interesting differences between two representations of problem solutions. This approach is restricted to domains in which students can construct formal representations of problem solutions. Once we have fine tuned this approach for the ER domain and established the range and limits of its educational value, we will be able to test the value of the approach in other domains that have this property.

5. Acknowledgments

We thank our colleagues at the Learning Research and Development Center of the University of Pittsburgh for their support, in particular Alan Lesgold for hosting the first author as a visiting scholar the second as his research associate, Sandy Katz for discussions about collaborative learning, and Kim Harrigal and Dan Jones for their assistance with the implementations. We also thank Jose G. Escamilla of ITESM for his advisorship of the first author, our domain expert Jose I. Icaza of ITESM, Moraima Campbell of ITESM for her support in user interface design, and the Center for Artificial Intelligence of ITESM for facilities for the research. During this research, Constantino-González was funded by ITESM Campus Laguna, and Suthers was funded by the Presidential Technology Initiative (while at LRDC) and by NSF's Learning and Intelligent Systems (while at the University of Hawai'i).

References

1. Abrami, P.C. and Bures, E.M. (1996) Computer Supported Collaborative Learning and Distance Education, Reviews of lead article. The American Journal of Distance Education 10(2): 37-42.
2. Ayala, G. and Yano, Y. (1996). Learner Models for Supporting Awareness and Collaboration in a CSCL Environment. ITS'96, Third International Conference of Intelligent Tutoring Systems, Montreal, June, pp. 158-167.
3. Baker, M. J. and Bielaczyc, K. (1995). Missed opportunities for learning in collaborative problem-solving interactions. *AI&ED 95.*
4. Baker, M.J. and Lund, K. (1996). Flexibly Structuring the Interaction in a CSCL environment., In EuroAIED.

In Intelligent Tutoring Systems, Proceedings of the 5th International Conference (ITS 2000), G. Gauthier, C. Frasson, & K. VanLehn (Eds.), pp. 325-333. Berlin: Springer-Verlag.

5. Batini, C., Ceri, S. and Navathe, S. B. (1992). Conceptual Database Design: An Entity-Relationship Approach. Benjamin/Cummings, Redwood City, California.
6. Batra, D. and Antony, S.R. (1994). Novice Errors in Conceptual Database Design. European Journal of Information Systems, Vol. 3, No. 1, pp. 57-69.
7. Brown, A. L. and Palincsar, A.S. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. Resnick (Ed.), Knowing, Learning and Instruction: Essays in Honor of Robert Glaser. Hillsdale, NJ: Lawrence Erlbaum Associates.
8. Chen, P. (1976). The Entity-Relationship Model - Toward a Unified View of Data, ACM Transactions on Database Systems, Vol. 1, No. 1, pp. 9-36.
9. Gokhale, A. A. (1995). Collaborative Learning Enhances Critical Thinking, Journal of Technology Education, 7(1), 1995.
10. Gordon, A. and Hall, L. (1998). A Collaborative Learning Environment for Data Modelling, American Association for Artificial Intelligence.
11. Ikeda, M., Go, S., & Mizoguchi, R. (1997, August 18-22). Opportunistic Group Formation. Paper presented at the AI-ED 97: World Conferennce on Artifical Intelligence in Education, Kobe, Japan.
12. Johnson, D.W., Johnson, R.T and Smith, K. A. (1997). Academic Controversy. Association of the Study of Higher Education.
13. Johnson, D.W., Johnson, R. T. and Smith, K.A. (1991). Increasing College Faculty Instructional Productivity, ASHE-ERIC Higher Education Reports.
14. McManus, M. M. and Aiken, R.M. (1995). Monitoring Computer Based Collaborative Problem Solving", Journal of Artificial Intelligence in Education , 6(4) , 308-336.
15. Okamoto, T., Inaba, A. & Hasaba, Y. (1995). The Intelligent Learning Support System on the Distributed Cooperative Evironment, Proceedings of Artificial Intelligence in Education, August, Washington, D.C., p. 588
16. Paolucci, M., Suthers, D., & Weiner, A. (1996). Automated advice-giving strategies for scientific inquiry. Intelligent Tutoring Systems, 3rd International Conference, Montreal, June 12-14, 1996.
17. Rabow, J., et al. (1994). Learning Through Discussion, Sage.
18. Robertson, J., Good, J. & Pain, H. (1998). BetterBlether: The Design and Evaluation of a Discussion Tool for Education, IJAIED.
19. Shanks, G. (1996). Conceptual Data Modelling: An Empirical Study of Expert and Novice Data Modellers. ACIS '96.
20. Soller, A., Goodman, B., Linton, F. & Gaimari, R. (1998). Promoting Effective Peer Interaction in an Intelligent Collaborative Learning System. Proceedings of the Fourth International Conference on Intelligent Tutoring Systems (ITS 98), San Antonio, Texas.
21. Suthers, D. (1993). Preferences for Model Selection in Explanation. Proceedings of the13th International Joint Conference on Artificial Intelligence (IJCAI-93). Chambery, France, August 1993, pp. 1208-1213.
22. Suthers, D. and Jones, D. (1997). An Architecture for Intelligent Collaborative Educational Systems, AI&ED97, Japan.
23. Webb, N. (1985). Student Interaction and learning in small groups: A research summary. Learning to Cooperate, Cooperating to Learn.
24. Webb and Palincsar, A. S. (1996). Group processes in the classroom. Handbook of Educational Psychology. D. Berlmer & R. Calfee Eds. Simon & Shuster Macmillan NY, 1996