Automated Coaching of Collaboration based on Workspace Analysis: Evaluation and Implications for Future Learning Environments

María de los Angeles Constantino-González
Dept. of Computer Sciences
ITESM Campus Laguna
Paseo del Tecnológico #751,
Col. Ampliación la Rosita
Torreón, Coah. 27250
México
aconstan@itesm.mx

Daniel D. Suthers
Dept. of Information and Computer Sciences
University of Hawai`i at Manoa
1680 East West Road, POST 309
Honolulu, HI 96822
USA
suthers@hawaii.edu

Abstract

This paper describes the design and evaluation of COLER, a computer mediated learning environment that includes a software coach to help students collaborate while solving Entity Relationship modeling problems. Unlike previous work generally emphasizing dialogue analysis or expert models, this work evaluates a new approach to supporting collaboration that identifies learning opportunities based on differences between problem solutions and tracking levels of participation. Results of an evaluation process explained below show that good quality advice can be obtained through these knowledge sources, although other knowledge sources may fill in gaps relative to the expert's performance. This work demonstrates how intelligent agents can produce reasonable collaboration advice in domains for which structured problem solutions exist by using a few basic knowledge sources. We discuss how COLER's design supports collaboration and implications for the design of future constructivist learning environments.

1. Introduction

Global changes in society and technology are demanding new skills and attitudes from today's workforce. Individuals should be able to share and negotiate their own knowledge and perspectives with others while working together to solve complex problems. Education must change from a transmission model of learning to a constructivist view of the learning process in order to support development of these skills. The learners' and professors' roles need to adjust accordingly [27]. Learners should participate actively in their learning, while professors should become learning facilitators who monitor students, guide participants in the application of collaborative skills and foster peer interaction by encouraging participation, discussion of viewpoints and resolution of conceptual disagreements. Collaborative learning is an appropriate instructional paradigm for this model of learning, and one for which positive outcomes have been reported, such as better performance, critical thinking and cooperative behavior ([15], [19], [29]).

Computer-mediated collaborative learning [23] is especially important in distance learning environments since remote learners may be isolated and have few opportunities to develop and practice collaborative skills. Abrami and Bures [1] state, "… with social and intellectual isolation, students may fail to develop and refine those cognitive and interpersonal skills increasingly necessary for business and professional careers." However, merely providing distance learners with a communication channel does not ensure that students will effectively collaborate with their teammates. A facilitator to monitor students' collaboration is necessary. Yet it is difficult for a human facilitator to track many teams working at different times with members located in different places. Therefore, there is growing interest in developing intelligent systems that support this facilitation process.

Our work provides a learning environment (COLER) in which the facilitator can interact with remote students, while being assisted by a computer coach that facilitates effective collaborative learning interactions. The main problems a collaboration coach has to solve are similar to the ones for individual coaching: when to intervene and what to say. Yet designing a coach that supports students' collaboration is a new challenge, since most prior work on coaching has focused on expert and student modeling [21]. In contrast, a collaboration coach has to monitor not only one student's activities, but also the teammates' activities, and should encourage interactions that influence individual learning and the development of collaborative skills, such as giving and receiving help and

feedback, and identifying and resolving conflicts or disagreements ([12], [20], [35]).

Several computer-mediated collaborative learning environments have been developed to support collaborative interaction [18]. Most of these systems, such as DEGREE [5], C-CHENE [4], the Group Leader Tutor [24], iDCLE's Expert System Coordinator [26], and BetterBlether [28], use restricted menu-driven or sentence-opener interfaces in order to understand students' interaction, and give guidance based on an ideal model of dialogue. Dialogue-based support provides several advantages, such as potential applicability to any subject matter area, automated interpretation of students' interactions, and restriction of discussion moves and learning interactions to those believed to be productive for learning. However, systems that require use of devices such as sentence openers present some disadvantages such as restricting the type of communicative acts, slowing the communication process, and misinterpreting students' dialogue when students use the interface buttons incorrectly. It would be advantageous to increase the repertoire of ways to provide automated support. One example is using action-based collaboration analysis [25], which monitors and analyzes moves of multiple users within a shared workspace. Another example is GRACILE [2] which gives help based on Vygostky's concept of the zone of proximal development.

Our work seeks to facilitate effective collaborative learning interactions with minimal reliance on restricted communication devices such as sentence openers. We focus particularly on helping students to recognize and resolve conflicts between their problem solutions, because these kinds of collaborative interactions are expected to lead to learning. In this paper, we describe the learning environment and evaluate the feasibility of generating advice based primarily on comparing students' individual and group solutions and on tracking student participation. We excluded the use of discourse models and comparison to expert solutions as a research strategy, in order to evaluate the value of the knowledge sources on which we focus. This strategy should not be interpreted as a denial of the importance of these other knowledge sources. Our approach is close in spirit to the action-based analysis of Mühlenbrock & Hoppe [25], but differs in that we monitor individual work in both private workspaces and shared workspaces to identify conflicts.

The paper is organized as follows. First we introduce the domain of Entity Relationship modeling and COLER, the learning environment within which the studies were undertaken. Then we describe the architecture of COLER's coach, and summarize our evaluations of the quality of COLER's advice, the roles of the knowledge sources in generating this advice, and students' opinions. We conclude with a discussion of the implications of this work for the next generation of online learning systems.

2. COLER

COLER (COllaborative Learning environment for Entity-Relationship modeling) is a Web-based collaborative learning environment in which students can solve conceptual database design problems while working synchronously in small groups at a distance. Conceptual database design requires analysts and database users to collaborate to produce a shared conceptual schema ([6], [16]). COLER gives instructors the facility to monitor students while they are collaborating on this task in real time and to give them advice. A personal coach, to support professor's facilitation, is included for each student within COLER. Each coach analyzes the state of interaction and recommends actions the student might take to improve his or her participation.
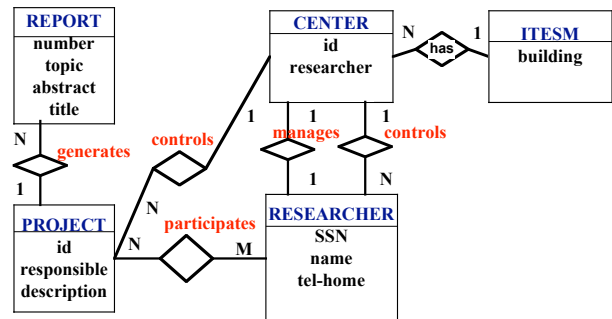

Figure 1: A student's entity-relationship diagram

The ER model is a diagram composed of a set of basic elements called entities, attributes and relationships. The Entity-Relation (ER) diagram notation used in this research is based on the Chen formalism [9], with IDEF1X notation [7] used to represent entities' attributes and reduce diagram space. An example student diagram is presented in Figure 1.

2.1 COLER's Functionality

The learning objectives for COLER are to improve students' performance in the application of the Entity-Relationship modeling formalism and to help students develop collaborative and critical thinking skills. Special emphasis is given to those collaborative skills associated with the analysis of different solutions and the expression and resolution of disagreements, such as Communication and Conflict Management skills [19]. COLER's aim is to take advantage of the positive social aspects of cognitive conflict, using the perception of differences as a motivating factor for discussion an their resolution as a means for learning ([13], [14]). COLER enables students to analyze alternatives, be aware of teammates' opinions, perceive differences in solutions, give and receive help
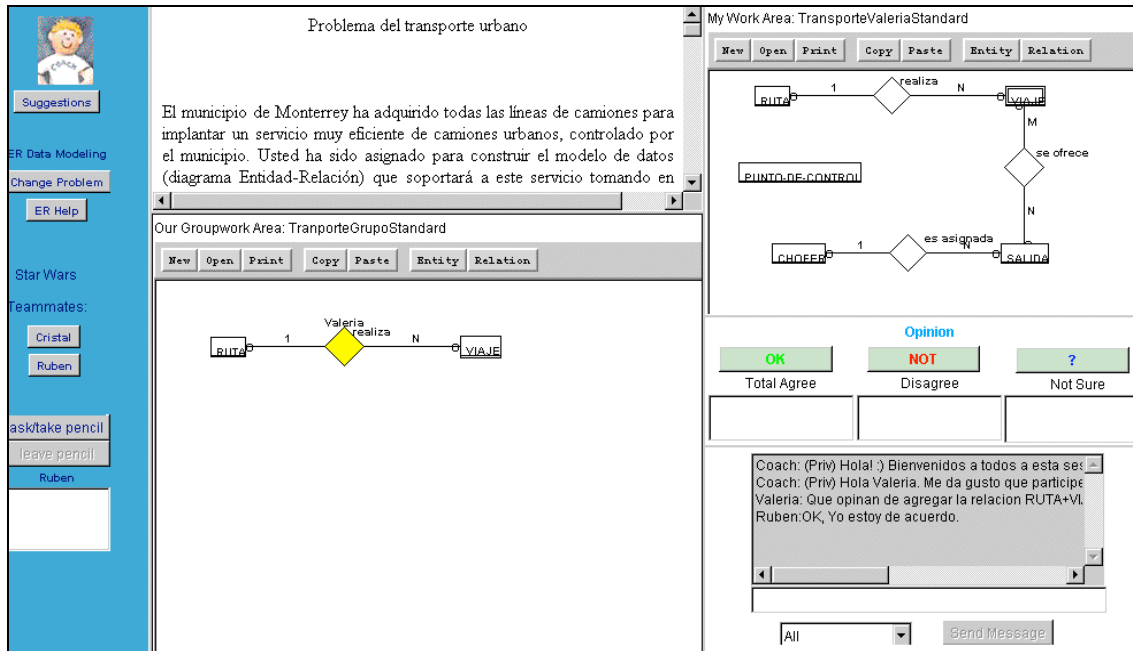
Figure 2: COLER collaborative student interface

and feedback and express disagreements. Students are encouraged to discuss their discrepancies and participate. They must solve the problem first alone and then in a group, which helps ensure individual participation and provides the raw materials for the proposed "negotiation of differences" paradigm of collaborative learning. Students' initial solutions also provide the coach with useful information to identify learning opportunities.

COLER enables instructors to setup some initial conditions and participate during the group interaction. Setup functions should be performed before students can work in a COLER collaborative session. They include defining teams (name and members) and creating the HTML pages of the database problems to solve. Additionally, the instructor must define the glossary of words for each problem, as well as review/update the value (weight) of typical ER differences between diagrams to indicate which ones are worth discussing.

After the instructor has finished all setup activities, students can launch COLER and log into the system. In an individual student session, a page containing the scenario of the selected problem and the specialized area for constructing ER diagrams is displayed This phase of individual problem solving is very important because it is expected that if students first try to solve the problem individually they are more likely to become aware of differences to be discussed. After every member of the team has finished his/her individual diagram, students can start working in groups. During the collaborative session, instructors can observe students' individual and collaborative progress and give students private or public advice through the chat window.

2.2. COLER's Interface

COLER's student group interface is shown in Figure 2. The *problem description window* (upper center) presents an entity-relationship modeling problem. Students construct their individual solutions in the *private workspace* (upper right). 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 the teammates already connected. Only the student 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 control and the students waiting for a turn. An *opinion panel* (middle right) contains two areas. The upper area contains three buttons that enable students express their opinion regarding the last object added to the shared area: **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 the chat area along with any justifications in order to correlate these opinion-expressing actions with the chronology of the chat discourse. The bottom area within this panel shows a

persistent summary of the teammates' opinions on a current issue by showing the teammates' name in the box below the opinion button the teammate selected (OK, NOT, ?). A *personal coach* (upper left) gives advice in the chat area based on 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 obtained by pressing the **SUGGESTIONS** button, which is disabled if the coach does not have any advice to offer.

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. Entities and attributes must be named with words in the Glossary of the selected problem to make students aware of the necessity of a common data dictionary and to make it easier for the coach to compare their solutions. The system doesn't attempt to do natural language understanding. Students can type what they want, but if a word used is not in the glossary, they are prompted to check the glossary and pick a synonym from there. 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.

Five formative empirical studies were conducted for COLER's design. Some of these studies were carried out on paper, while others used COLER's initial interface. We found that it is important to have the problem statement available as part of the interface, since students usually read the problem while they are constructing a solution and point out a specific part of the problem when they discuss a specific issue. Although in the paper-based studies the group solution was usually constructed by one student, in the online studies more than one student generally wanted to contribute to the group diagram. These facts corroborated the idea of defining private and shared windows. Verbal communication during the studies highlighted the importance of providing students with a communication channel, such as the chat facility. Problems with coordination and conflict of actions in the workspace motivated the addition of access control with the "pencil". Other problems in students' participation and attention to the contributions of others motivated the use of Opinion buttons and confirmed the potential utility of a coach to help students to collaborate.

2.3. COLER's Implementation

COLER's implementation is based on an architecture for intelligent collaborative learning systems [31] originally used for the implementation of the Belvedere
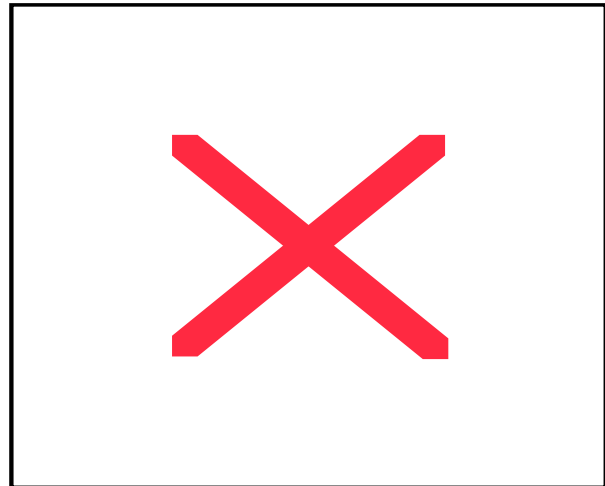

Figure 3. COLER's architecture

software for collaborative critical inquiry [32]. Belvedere was selected to implement COLER primarily because it combines collaborative learning, coached apprenticeship and problem-based learning paradigms. COLER's architecture is shown in Figure 3. The light objects indicate modules that are Belvedere extensions. The dark objects are new modules. Applets are in charge of the different functions of the system, such as chat, floor control, voting, private and shared ER modeling. 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 of the shared workspace, chat window, and opinion and team panels.

Each student's client contains a personal coach. This personal coach is a Java thread that monitors participation, identifies and evaluates differences between diagrams and encourages students to discuss them.

3. COLER's Coach

According to Collins, Brown & Holum [10], *coaching* is a technique in which the instructor observes students and provides hints, help and feedback while they try to complete a task. Since students often miss learning opportunities and get stuck on a certain level of proficiency, a coach can make students aware of further possibilities, provide unobtrusive assistance and create potential learning experiences that will improve individual's development. COLER's coach is a pedagogical agent to facilitate collaboration. It does not tutor Entity Relationship modeling, but encourages
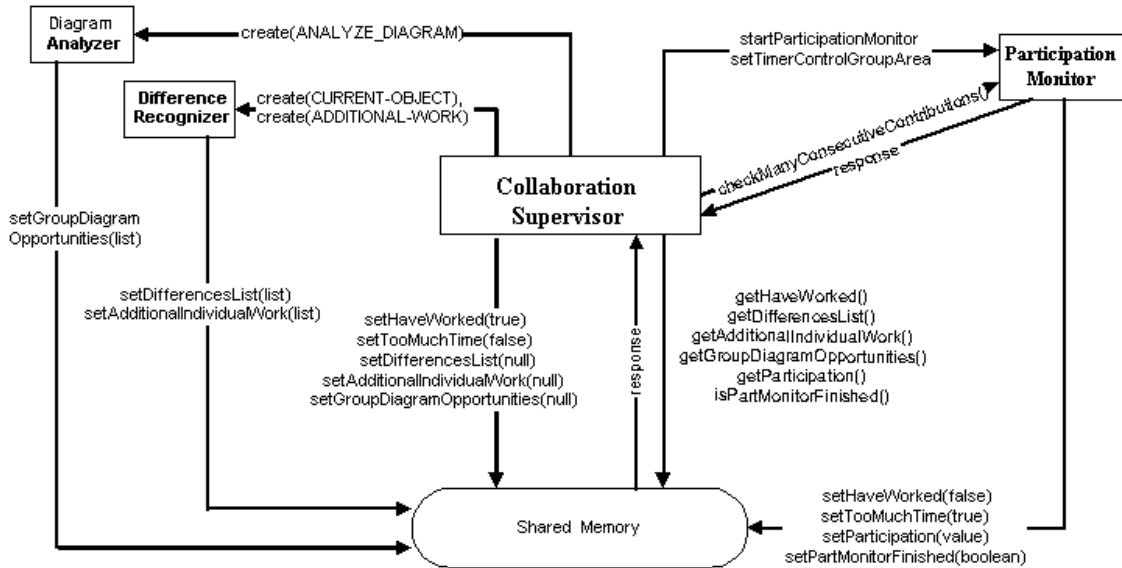
Figure 5. Communication between COLER coach Modules

students to discuss and participate during collaborative problem solving. The coach is implemented as a personal assistant for each student.

3.1. Coach's Advice

The coach's goal is to promote group-learning interactions and maintain balanced participation. The coach helps to prevent missed opportunities for collaborative learning [3] by monitoring students' participation and recognizing differences between students' individual and group solutions. When relevant opportunities for learning are found, the coach tries to guide students to practice collaborative skills. COLER's advice is expressed as suggestions or questions, so students should feel free to follow the advice or discard it when they believe it to be inappropriate. Advice types and categories were defined based on the collaborative learning literature and Wizard of Oz studies in which the human expert coached through the chat interface: see [11]. The present version of COLER includes seven advice *categories*. The first two categories, *Discussion* (in chat) and *Participation* (in the group workspace), are the main categories related to coaching collaboration. *Feedback* messages are related to student's pressing of COLER opinion buttons. The *ER Modeling* category includes suggestions related to some common errors in the domain. The *Self-Reflection* category consists of suggestions to think about a problem or situation. COLER also uses *welcoming* and *goodbye* messages. *Types* of advice were defined and classified within these categories.

3.2. Coach's Architecture

The current version of the personal coach is implemented as a local component of the COLER application. The coach relies completely on information that is local to the application. There is no direct communication between different students' personal coaching agents, although group parameters are accessible through the local copy of the shared workspace. The coach involves different modules, each implemented as a Java thread. The types of messages they use in the communication are presented in Figure 5. The expertise contributed by each module is summarized below.

The *Diagram Analyzer* recognizes some common errors in the group ER diagram based on their structural and categorical characteristics. Detection of these problems was included to see how they could be used to encourage students' participation.

The *Difference Recognizer* detects opportunities for students to collaborate by finding significant differences between individual and group ER diagrams. COLER mainly recognizes differences by identifying a number of significant syntactic dissimilarities between individual and group ER diagrams. A weight was assigned to each difference depending on its impact, and is used to decide when to give advice. A glossary of terms based on the significant nouns mentioned in the problem description is used to enable difference detection. The Difference Recognizer undertakes subgraph matching between the private and group ER diagrams for the purpose of identifying differences. Matching can either find differences specifically related to the currently added object (e.g. missing entity, extra attribute), or find all "extra work" that the student can contribute to the group.

The *Participation Monitor* attends to the activity in the group diagram. It detects time-triggered events, such as inactivity in the group area or the coached student having the control of the group area (pencil) for a long time. Group diagram events, such as addition of an object, are also detected so it can monitor whether each student is participating too much or too little.

The Diagram Analyzer, Difference Recognizer and Participation Monitor communicate their results to the *Collaboration Supervisor* via a shared memory. The Collaboration Supervisor maintains an internal model of the environment which includes the current group and individual diagrams, "the coached student" and team members' levels of participation, advice types, advice patterns and advice history, current received and given feedback, session phase and session start time. The Collaboration Supervisor operates in two phases: *Advice Generation* and *Advice Selection*, described in greater detail below.

3.3. Generating Advice

The Advice Generation phase of the Collaboration Supervisor module uses event-driven reasoning. Given an event, the Collaboration Supervisor identifies the event type, analyzes the situation and then decides what kinds of advice to give. Three main types of events are attended: (a) Time-triggered events, such as inactivity in the group diagram (b) Group and individual diagram events, for example, the addition, change or removal of an object, and (c) Voting events, such as the receiving and giving of feedback. An AND/OR decision tree is associated with each event type. This tree generates different advice types given an event type. Every branch of the tree represents a possible set of suggested advice. Several suggestions might be generated for any given event because several leaves may be reached at once via the "and" arc of the tree. Also, many of the leaves of each tree generate multiple advice, and trees for different events may be invoked at the same time.

3.4. Selecting Advice

Advice Selection is the second phase of the Collaboration Supervisor Module. Advice selection is sensitive to the relative importance of different forms of advice and to the context of the advice (problem solving and advice giving history). Six *control strategies* were specified to control selection and timing of advice: Preferences, Collaborative Session Phases, Discussion Encouragement Intensity, Participation Balance, Time on Task and Waiting for Feedback. Most of these strategies, except the first one, are employed to filter advice during the generation process: see [11]. Three kinds of

preferences were defined: New Advice (don't repeat advice type during the session), Many Instances (prefer advice of a type that applies more than once) and Category preferences (e.g. prefer Discussion, Participation, or Feedback advice). The preferences in use can change during the group session according to the group's performance or the current Collaborative Session Phase. The advice selection process involves two steps. First, select an advice pattern from the advice types generated by each one of the leaves. Second, select an advice from the resultant advice set using a preference-based sort algorithm [30] to choose between multiple advice instances. From the sorted list, the coach gives the more preferred advice. The others are stored in a list to be given on demand.

4. Evaluation

The evaluation reported here assesses the quality of advice generation and selection algorithms, the contributions of knowledge sources in the generation of reasonable advice and student's opinions of COLER.

4.1. Summary of Method and Procedure

This laboratory evaluation of COLER involved participants who had taken or were taking a database course. Our domain expert, a computer science professor, was also present in two sessions. Five sessions were conducted to generate data and scenarios for the different types of evaluations. In each of these sessions, three students were presented with a simple database design problem. They first solved the problem individually, and then convened to construct a group solution. Students and coach activities were recorded in a log file. The two sessions in which the Expert was present were used for preliminary evaluation, detecting some problems in COLER's user interface and coach algorithms. The last three sessions, in which the expert was not present, were used to evaluate COLER's algorithms and the quality of its advice, as described below. For each student of each of the last three sessions, two documents were generated for the Expert's Evaluation: the Environment Document and the Advice Document. Both documents describe the chronological sequence of events of the collaborative session in reference to a specific student, and the context of each event (current state of the environment). The expert reviewed each event and indicated what advice (if any) he would give in response to that event. He also rank ordered the advice types available to COLER and categorized them as "Worth Saying", "So-So," and "Not Worth Saying." All of this was done without knowledge of the advice that COLER gave.

4.2. Results

Advice generated by COLER included 34 Participation, 23 discussion, 6 Self-Reflection and Discrepancy, and 9 Feedback advice. Our evaluation of this advice and various aspects of its generation and selection follow below.

The *overall knowledge* available to COLER was evaluated by comparing expert and COLER advice for each situation, with 67% of the advice given by the expert not given by the coach. Thus, as expected the expert has a greater repertoire of advice, although COLER's limited knowledge sources produced the same advice as the expert in 33% of the situations. The distribution of the missing advice is: 69% would require new advice types and new branches in the AND/OR decision tree (New), 21% involved situations already considered in the AND/OR tree but requiring that new advice types be attached to them (Considered), and 10% involved advice that COLER could give with minor adjustments to parameters (Parametric). According to the results, some existing advice types need to be extended to mention a specific context. The findings also suggested situations in which a new "Self-Reflection" advice type could be given. The expert also suggested social interactions such as thanking the student for listening to advice, and otherwise commenting on student actions. However, these kinds of messages are not closely related to the main objective of the coach. Rather the coach is trying to assist students by helping them to visualize differences and encouraging them to leverage learning opportunities provided by the online group.

Advice quality was evaluated by using the expert's classification of the advice available to COLER into "Worth saying," "So-So" and "Not Worth Saying." Results showed that 73% of the advice generated by COLER was worth saying, 7% was "so-so" and 20% was not worth saying. Some reasons for "Not Worth Saying" advice are change in conditions making the advice obsolete and failure to match entities due to spelling errors and unidentified synonyms. These problems can be addressed by checking conditions before giving computed advice and by improving the lexical knowledge of the system, respectively.

The *Advice Selection* module was evaluated by analyzing events in which (1) several candidate advice with different rankings exist, and (2) some advice was suggested by the Expert. To evaluate the selection algorithm independently of the generation algorithm, a ranking of the candidate advice generated by COLER for each situation was computed based on the Expert ranking of all advice with respect to that situation. Selection among several advice types was needed only a few times in this study. There were 2.33 average selections per session, each selecting between an average of 4.78 generated advice items. The Euclidian distance obtained was 0.9, i.e., less than one disagreement in ranking per selection event. This seems adequate although leaving room for minor improvements.

The *contribution of knowledge sources* to the generation of advice was evaluated by "ablating" COLER analytically, i.e., identifying the advice that relied on the knowledge source and hence would be lost if the knowledge source were removed. We focused on the advice that the expert ranked as "Worth Giving." This analysis used the Environment and Advice documents to identify the situations in which COLER gave advice and the rank the Expert assigned to this advice, and the AND/OR trees to identify the type of knowledge used in each situation. The contribution of knowledge sources to generation of advice judged by the expert to be worthwhile as follows: Tracking voting and feedback timeout (49%), Participation balance (48%), Significant differences and problem glossary (41%), Time on task (40%), Chat tracking (37%), Discussion encouragement intensity parameters (29%), Category and sort preferences (22%), Pencil tracking (14%) and Problems in ER diagrams quality (2%). Some knowledge sources were used to generate different categories of advice (hence the percentages reported above sum to greater than 100) while others were more marginal and only were used in a specific advice category. The generation of worthwhile advice often required the conjunction of several types of knowledge (e.g. Significant Differences and Problem Glossary, Participation Balance) and confirmed the hypothesis that knowledge on problem solving activity could be used to generate reasonable collaboration advice. The knowledge about problems in quality of ER diagrams was used very little in this study since the coach's primary goal is promote discussion and participation instead of teaching ER modeling.

A questionnaire was designed for *students' evaluation* of COLER's behavior. These questionnaires were administered after the collaborative problem-solving phase. Students indicated that COLER did help them to collaborate. They mentioned that the chat is an important tool for communication and that giving an opinion (OK, NOT, ?) when an object was added motivates them to reach agreement before moving on. However, they also recommend various improvements, such as making the system faster, having the facility to comment about an object different to the last one added, sending chat messages by pressing ENTER, including colors in chat, making the work areas bigger and including some common buttons and commands used in other applications. General comments about the system indicate that students enjoyed collaborating through COLER's interface and this is a good way to practice collaboration.

5. Contributions and Future Work

This work contributes to a research agenda that seeks to characterize the knowledge needed to facilitate collaborative learning processes. The present focus has been on determining how much leverage can be obtained by a basic ability to detect semantically interesting differences between representations of two problem solutions, together with simple tracking of individuals' levels of participation (e.g. contributions in the shared Workspace) and feedback given (e.g. opinion buttons). The study showed that reasonable collaboration advice can be generated without the need for expert solutions or discourse understanding, although the addition of these knowledge sources would improve the quality and range of advice generated and selected by the system (at the cost of additional knowledge engineering and system complexity). Results indicate that COLER is a viable advisor, albeit different in style from our expert.

One limitation of this research is that only one expert evaluated the system. Testing the approach in other problem domains would also strengthen the generality of the conclusions. Although COLER is presently limited to Entity Relationship (ER) modeling, its long-term goal is to support other domains in which problem solutions can be expressed in structured representations that can be compared to identify significant differences, such as finite automata modeling, object oriented modeling and systems dynamics. The following process would be required to apply the coach's design to other areas: (1) identifying how diagram objects can be compared (e.g. lexical units in finite automata), (2) identifying the significant differences in the domain (e.g. missing link, extra link, missing a node, extra node), (3) modifying the Difference Recognizer accordingly, (4) adjusting the differences' weights according to their relevance, (5) reviewing participation and discussion parameter settings and (6) adjusting minor portions of the Advice Generator. Several of the coach's modules can be reused.

In this work, a personal assistant in each client viewed a given student's private workspace and the shared workspace in order to help prevent missed opportunities for collaborative learning. Future work may investigate a single *global* coach endowed with the ability to inspect all students' private workspaces as well as the shared workspace. Such a coach would be able to identify conflicts between solutions in private workspaces and encourage the students to share the relevant part of their solutions, thereby *creating* conflict opportunities for collaborative learning. Little additional knowledge engineering would be required.

5.1. Implications for Next Generation Learning Platforms

As online learning grows in popularity, the need to monitor both group interaction and individual work will increase. This monitoring could be difficult for a human facilitator, especially given that large numbers of students may be interacting at many different times of day. Therefore, our demonstration that reasonable guidance can be generated by basic monitoring of participation and individual and group workspaces has significant practical applications for augmenting human coaching in the learning platforms of the future. We expect that the techniques demonstrated in this paper should generalize to any learning applications that require students to generate and compare structured problem solutions.

In addition to the automated coaching facility, COLER demonstrates several integrated design elements that facilitate remote collaboration and to promote students' participation and discussion of discrepancies. We believe that new architectures for learning should provide similar elements to enable efficient collaboration and learning facilitation. These design elements include the chat facility, the shared workspace with a floor control policy (e.g. pencil), and the simultaneous presence of an individual workspace so that students can perceive differences between their own solution and the group's solution. The mandatory use of buttons **OK, NOT** and **?** involves students in the collaborative task and motivates them to participate and express their opinion. COLER's interface allows the professor to interact with students during the collaboration process either at the same time or at a different time than a software coach.

This design stands in contrast to many commercial course managements systems, where discussion tools (whether chat or threaded discussion), whiteboards, and areas for posting student work are not well integrated, and may not even be viewable together, presenting an impediment to effective collaboration. Structuring of collaborative interactions (e.g., the Opinion buttons) and coaching of the same are both entirely absent from commercial environments.

Yet COLER is far from sufficient in itself. Many improvements and additional elements should be included in a complete learning platform. We will only attempt to mention some that are directly related to collaboration and to coaching of collaboration.

In this work we focused on demonstrating the value of tracking the contents and activity in private and shared workspaces. Other knowledge sources are available and could improve the quality of coaching if properly employed. Extensions that use models of discourse and natural language understanding to track the extent to which students are discussing and resolving differences in

the chat medium can be envisioned. A greater investment in domain-specific knowledge would enable the coach to compare student solutions to expert solutions in order to (a) guide advice selection (e.g., encourage students to share solutions that are correct), and (b) comment directly on the correctness of solutions in the manner of traditional intelligent tutoring systems or Web-based adaptive systems, such as the ELM-ART Lisp tutor [36] or the German tutor presented by Heift & Nicholson [17].

COLER could be extended to include a model of specific roles, enforcing roles with the interface, and coaching these roles [8]. With or without roles, visual indicators of each teammate's behavior (e.g., relative levels of participation) in the student and professor's interfaces could help students evaluate how well they are collaborating and the professor to evaluate group interaction.

We noted that the statement of the problem (as well as the group workspace) is often referenced by students in their discussions. Students would also benefit from being able to view and discuss different representations of the problem solution. For instance, the relational model (tables) could be displayed and manipulated in addition to the ER diagrams. These are examples of how learning discourse is often *artifact-centered*, a fact that has implications for the design of online learning platforms. Stronger integration between the discourse medium (e.g., chat) and other representations is needed to support deictic references, improve the coherence of discourse, and encourage convergence on solutions ([33], [34]).

In general, we see a trend in learning technology research and development towards providing learning environments that integrate multiple functionalities, such as tools for argumentation and discourse with modeling and simulation tools and intelligent coaches [22], in a manner driven by an understanding of the multi-faceted needs of learners.

6. Acknowledgments

7. References

[1] Abrami, P.C. and Bures, E.M. *Computer Supported Collaborative Learning and Distance Education, Reviews of lead article*. The American Journal of Distance Education, 1996, 10(2): 37-42.

[2] Ayala, G. and Yano, Y. Learner models for supporting awareness and collaboration in a CSCL environment. *Proceedings of the Third International Conference of Intelligent Tutoring Systems (ITS'96)* Montreal, Canada, 1996 pp. 158-167.

[3] Baker, M.J. and Bielaczyc, K. Missed Opportunities for Learning in Collaborative Problem-solving Interactions. In Greer, J. (Ed.) *Proceedings of the AI-ED 95–World Conference on Artificial Intelligence in Education,* Washington, DC, August 16-19. Association for the Advancement of Computing in Education, 1995, pp. 210–217.

[4] Baker, M.J. and Lund, K. Flexibly structuring the interaction in a CSCL environment. In P. Brna, A. Paiva & J. Self (Eds), *Proceedings of the European Conference on Artificial Intelligence in Education.* Lisbon, Portugal, Sept. 20 - Oct. 2, 1996, pp. 401-407.

[5] Barros, B. and Verdejo, M. F. Analysing student interaction processes in order to improve collaboration. The DEGREE approach. *International Journal of Artificial Intelligence in Education*, 2000, 11: 221-241.

[6] Batini, C., Ceri, S. and Navathe, S. B. *Conceptual Database Design: An Entity-Relationship Approach*. Benjamin/Cummings, Redwood City, California, 1992.

[7] Bruce, T.A. *Designing Quality Databases with IDEF1X Information Models*. Dorset House, New York, 1992.

[8] Burton, M., Brna, P, and Treasure-Jones, T. Splitting the Collaborative Atom: how to Support Learning about Collaboration. In B. du Boulay and R. Mizoguchi (Eds.) *Artificial Intelligence in Education*, IOS Press, 1997, pp. 135-142.

[9] Chen, P. The Entity-Relationship Model - Toward a Unified View of Data, *ACM Transactions on Database Systems*, 1976, 1(1): 9-36.

[10] Collins, A., Brown, J. S., and Holum, A. Cognitive apprenticeship: Making thinking visible. *American Educator,* Winter, 1991, pp. 6-46.

[11] Constantino-González, M. A. *A Computer Coach to Support Collaboration in a Web-based Synchronous Collaborative Learning Environment*. Unpublished dissertation, ITESM (Instituto Tecnológico y de Estudios Superiores de Monterrey), México, 2000.

[12] Dillenbourg, P., Baker, M., Blaye, A. and O'Malley, C. The Evolution of Research on Collaborative Learning. In E. Spada & P. Reiman (Eds). *Learning in Humans and Machine: Towards an interdisciplinary learning science*, Oxford: Elsevier, 1996, pp. 189-211.

[13] Doise, W., and Mugny, G. The social development of the intellect. *International Series in Experimental Social Psychology*, 10, Pergamon Press, 1984.

[14] Festinger, L. *A theory of cognitive dissonance*. Stanford University Press, 1957.

[15] Gokhale, A. A. (1995). Collaborative Learning Enhances Critical Thinking, *Journal of Technology Education,* 7(1), 1995.

[16] Gordon, A., and Hall, L. A collaborative learning environment for data modelling. *Proceedings of the 1998 Florida Artificial Intelligence Research Symposium*, Sanibael Island, FA:AAAI Press, 1998, pp. 158-162.

[17] Heift, T. and Nicholson, D. Web Delivery of Adaptive and Interactive Language Tutoring. *International Journal of Artificial Intelligence in Education*, 2001, 12.

[18] Jermann, P., Soller, A., & Muehlenbrock, M. From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. *European Perspectives on Computer-Supported Collaborative Learning: Proceedings of the First European Conference on Computer-Supported Collaborative Learning,* Maastricht, The Netherlands, March 22-24, 2001, pp. 324-331.

[19] Johnson, D.W. and Johnson, R.T. *Learning Together and Alone*, Englewood Cliffs, NJ: Prentice Hall, 1994.

[20] Johnson, D.W., Johnson, R. T. and Smith, K.A. *Increasing College Faculty Instructional Productivity*, ASHE-ERIC Higher Education Report No. 4. School of Education and Human Development, George Washington University, 1991.

[21] Katz, S. and O'Donnell, G. The Cognitive Skill of Coaching Collaboration. *In Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 Conference*, C. Hoadley & J. Roschelle (Eds.) Dec. 12-15, Stanford University, Palo Alto, California. Mahwah, NJ: Lawrence Erlbaum Associates, 1999, pp. 291-299. [Available at http://kn.cilt.org/cscl99/A36/A36.HTM].

[22] Koedinger, K., Suthers, D., and Forbus, K. Component-based Construction of a Science Learning Space. *International Journal of Artificial Intelligence in Education*, 1999, 10: 292-313. Available: http://cbl.leeds.ac.uk/ijaied/home.html

[23] Koschmann, T. Paradigm Shifts and Instructional Technology: An Introduction. *CSCL: Theory and Practice of an Emerging Paradigm.* Lawrence Erlbaum Associates, Publishers. Malwa, N.J. 1996, pp. 1-23.

[24] McManus, M. M. and Aiken, R.M. Monitoring Computer Based Collaborative Problem Solving, *Journal of Artificial Intelligence in Education* , 1995, 6(4) , 308-336.

[25] Mühlenbrock, M., and Hoppe, U. Computer Supported Interaction Analysis of Group Problem Solving. In C. Hoadley & J. Roschelle (Eds.) *Proceedings of the Computer Support for Collaborative Learning (CSCL-99) Conference*, Stanford University, NJ: Lawrence Erlbaum Associates, Palo Alto, Dec. 12-15, 1999, pp. 398-405.

[26] Okamoto, T., Inaba, A. and Hasaba, Y. The Intelligent Learning Support System on the Distributed Cooperative Environment, In Greer, J. (Ed.) *Proceedings of Artificial Intelligence in Education AI-ED'95,* Washington, D.C., August 16-19, 1995, 588.

[27] Reigeluth, C. The imperative for Systemic Change. In Charles M. Reigeluth & R.J. Garfinkle (Eds.) *Systemic Change in Education.* Educational Technology Publications. Englewood. Cliffs. N.J. 1994, pp. 1-6.

[28] Robertson, J., Good, J. and Pain, H. BetterBlether: A Computer Based Educational Communication Tool. In G. Ayala (Ed.) *Proceedings of Workshop "Current Trends and Applications of Artificial Intelligence in Education" at the 4th World Congress on Expert Systems*, Mexico City, Mexico, March 16, ITESM, 1998, pp. 90-97.

[29] Slavin, R. E. *Cooperative Learning.* Allyn and Bacon, 2nd. Edition, 1995.

[30] Suthers, D. Preferences for Model Selection in Explanation. *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93).* Chambery, France, August, 1993, pp. 1208-1213.

[31] Suthers, D.D. and Jones, D. An Architecture for Intelligent Collaborative Educational Systems.In B. D. Boulay and R. Mizoguchi (Eds.) A*rtificial Intelligence in Education: Knowledge and Media in Learning Systems. (Proceedings of AI-ED'97, 8th World Conference on Artificial Intelligence in Education),* Kobe, Japan, Amsterdam: IOS, August 18-22, 1997, pp. 55-62.

[32] Suthers, D, Connelly, J., Lesgold, A., Paolucci, M., Toth, E., Toth, J., and Weiner, A. Representational and Advisory Guidance for Students Learning Scientific Inquiry. In K. D. Forbus and P. J. Feltovich (Eds.) *Smart Machines in Education: The Coming Revolution in Educational Technology.* Menlo Park: AAAI Press, 2001, pp. 7-35.

[33] Suthers, D. Collaborative Representations: Supporting Face-to-Face and Online Knowledge-building Discourse. *Proceedings of the 34th Hawai`i International Conference on the System Sciences (HICSS-34),* Maui, Hawai`i (CD-ROM), Institute of Electrical and Electronics Engineers, Inc. (IEEE), January 3-6, 2001.

[34] Suthers, D. & Xu, J. Kukakuka: An Online Environment for Artifact-Centered Discourse, *Education Track of the Eleventh World Wide Web Conference (WWW 2002),* Honolulu, May 7-11, 2002, pp.472-480. Available: http://www2002.org/CDROM/alternate/252/index.html

[35] Webb, N. and Palincsar, A. S. Group processes in the classroom. *Handbook of Educational Psychology.* D. Berlmer & R. Calfee (Eds.) Simon & Shuster Macmillan NY, 1996.

[36] Weber, G. and Brusilovsky, P. ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education*, 2001, 12, to appear.