

I-MINDS: A Multiagent System for Intelligent Computer-Supported Collaborative Learning and Classroom Management

Leen-Kiat Soh*, Nobel Khandaker, and Hong Jiang

Department of Computer Science and Engineering

*National Center for Information Technology in Education**

University of Nebraska

256 Avery Hall, Lincoln NE 68588-0115 USA

{lksoh, knobel, jiang}@cse.unl.edu

Abstract. I-MINDS, which stands for Intelligence Multiagent Infrastructure for Distributed Systems in Education, provides a computer-supported collaborative learning (CSCL) infrastructure and environment for learners in synchronous learning and classroom management applications for instructors, for large classroom or distance education situations. For supporting the instructor, I-MINDS includes a teacher agent. The teacher agent allows the instructor to interact with students, manage Q&A sessions, administer quizzes, post evaluations, form groups and monitor individual and group performances. For computer-supported collaborative learning, I-MINDS provides a student agent for each student. Each student agent monitors and models its user and carries out group formation with other student agents behind-the-scene. I-MINDS provides standard online collaborative features such as chat rooms and whiteboards and implements a structured cooperative learning mechanism called Jigsaw. The system also incorporates multiagent coalition formation algorithms for forming peer groups. I-MINDS also provides a group agent for each peer group that monitors and evaluates its assigned peer group. I-MINDS has been deployed and evaluated in an introductory computer science course (CS1). Results show that intelligent tools such as I-MINDS can be used in a real-time environment to support student cooperative learning activities, and also as a testbed to collect instructional or pedagogical data for better understanding of student collaborative learning. This paper describes the I-MINDS framework, implementation, deployment, and test results comprehensively.

Keywords. Computer-Supported Collaborative learning, Multiagent System

Students in large or distance classrooms often do not enjoy the same level and quality of student-to-student and student-to-instructor interactions during the class (synchronously) or after the class (asynchronously) as in smaller classrooms. Such interactions are, however, important to foster group-based learning and student affinity in a class. To improve such interactions, educators have turned to technology such as personal response systems, online chat rooms, teleconferencing tools, and computer-supported collaborative learning (CSCL) environments. However, most CSCL systems (e.g., Caverly and MacDonald 2002; Betbeder et al. 2003; Chan et al. 2003; Chang et al. 2003; Salcedo et al. 2003) do not adapt to diverse student and instructor needs and behaviors: they are passive participants in the process and students and instructors often resort to other modes of interactions (e-mail, phone, or face-to-face meetings) in order to address individual problems. Further, the advent of Internet and multimedia technology has meant potentially drastic changes in the teaching

and learning process from the traditional classroom setting to a more geographically distributed, virtual but still interactive one. Hence, there are both motivation and opportunity to develop *intelligent* CSCL systems.

In our research, an intelligent CSCL system is defined as one that is capable of enabling flexible behaviors and machine learning capabilities. Flexible behaviors include reactivity, proactiveness, and social ability, using the definition from the intelligent agent research community (Wooldridge & Jennings 1995). Adopted for the CSCL application, a system or its components should be able to (1) perceive its environment and respond in a timely fashion to changes that occur in it, (2) exhibit goal-directed behaviors by taking the initiative, and (3) interact with other agents and human users. On the aspect of machine learning, an intelligent CSCL system should be able to improve its performance over time, adapting to instructor and students needs through its interactions with them.

Given the above, I-MINDS, which stands for Intelligent Multiagent Infrastructure for Distributed Systems in Education, has been developed. I-MINDS consists of a set of intelligent agents. Among their many capabilities, an I-MINDS teacher agent evaluates and categorizes questions asked by students, profiles students based on the quality of their questions, and learns from the instructor to better evaluate the quality of questions; an I-MINDS student agent evaluates and forms a peer group (“buddy group”) for the student that it serves; and an I-MINDS group agent oversees and supports group activities among students. These agents also communicate and exchange information among themselves behind-the-scenes to facilitate student and group profiling. In addition to standard online collaborative features such as chat rooms and whiteboards, I-MINDS also implements a structured cooperative learning mechanism called Jigsaw (Clarke 1994) and incorporates an innovative multiagent coalition formation algorithm based on an auction protocol. Agents in I-MINDS are capable of machine learning. For example, a teacher agent learns to weight keywords by observing how the teacher chooses to respond to questions posed by the students, a group agent learns to put more compatible students in peer groups for cooperative activities, and so on.

I-MINDS was deployed and evaluated in actual classrooms in 2005 at the Department of Computer Science and Engineering of the University of Nebraska. For the evaluation, a control-treatment protocol was used to study the use of I-MINDS in place of face-to-face collaboration among students in weekly laboratory sessions of an introductory computer science course (CS1). Results showed that I-MINDS could support cooperative learning actively and effectively.

This paper is a comprehensive description of the I-MINDS project with extended discussions on the deployment and evaluation of I-MINDS. It describes a CSCL software infrastructure that is capable of monitoring and tracking both students and teacher activities, and making decisions to support the users. The infrastructure is also capable of machine learning, allowing the agents to improve their performance over time or to adapt to individual user behaviors. This paper further describes the I-MINDS environment such as the features for student collaborations and classroom managements, and reports on the deployment results of I-MINDS. Our work contributes to the area of CSCL in two ways. First, we have designed a multiagent framework where autonomous agents act to serve their human users as well as on their own to support learning among students and manage the classroom for the instructor. We have also designed a learning-based coalition formation algorithm, based on which student groups are formed, that could be used by other CSCL systems. Second, we have implemented a multiagent system (i.e., I-MINDS) that, as a distance learning tool, has comparable results in student performance with traditional face-to-face classrooms.

Previous work of I-MINDS has been reported in conference proceedings and as book chapters. In the early phase of the research and development of I-MINDS, agent-centric discussions on I-

MINDS can be found in (Soh et al. 2004a) and discussions on the system-centric distributed processing techniques can be found in (Liu et al. 2003a, 2003b). As the I-MINDS project entered its deployment and evaluation phases, the use of multiagent intelligence was proposed and described in (Zhang et al. 2005; Soh et al. 2006a), multiagent coalition formation algorithms were addressed in (Soh et al. 2006b), the incorporation of structured cooperative learning paradigm into I-MINDS was proposed and described in (Soh 2004, Soh et al. 2005b), and pedagogical studies were reported in (Soh et al. 2004b, 2005a; Soh 2006). Detailed design and implementation of the I-MINDS software can be found in several Master's theses and project reports: the multimedia support module (Vemuri 2003), the intelligent module (Namala 2004), the overall system design (Zhang 2004), and the coalition formation algorithm and structured cooperative learning study (Khandaker 2005).

METHODOLOGY

In this section, we present our approach to realizing the I-MINDS architecture. First, we present the theoretical framework that describes the principles that guides our design of I-MINDS. Then we describe the I-MINDS architecture, where we also describe how agents communicate and collaborate. After the description of the architecture, we provide the details of the agents in I-MINDS. Then we present VALCAM—an algorithm for forming learner groups and the Jigsaw learning model (Aronson et al. 1978)—to form structured cooperative learning groups. Finally, we present an example of use for I-MINDS.

Theoretical Framework

The theoretical framework of the I-MINDS system is based on three fundamental principles.

- **Principle 1.** First, we want to build a CSCL system that is responsive, flexible, distributed, and adaptive to individual student behaviors. The need for a flexible and adaptive framework arises from the diversity in the capability and motivations of the students who participate in a typical CSCL environment. A distributed framework would distribute the computational resource requirements and as a result make the system more responsive and fault tolerant.
- **Principle 2.** Second, we want to build a CSCL system that is able to evolve over time in terms of its pedagogical knowledge, student and even group modeling, and performance in decision support. Human users vary in their skill level, motivation, knowledge, experience, and so on. Thus, a non-adaptive solution or framework for CSCL that works for one type of students may not work for the others.
- **Principle 3.** Third, we want to build a CSCL system that is able to form effective student learning groups on its own. An automated learner group formation capability would make it easy for the instructor to conduct group activities in a large classroom.

In accordance to Principle 1, our solution is to use a multiagent system. A multiagent system or MAS can be defined as a set of intelligent agents that work and communicate together to achieve individual and common goals. According to (Huhns et al. 2000), a MAS *is the best way to design distributed computing systems*. In our multiagent solution framework for CSCL, we assign an intelligent agent to each learner. This agent works with and helps that learner to achieve his or her personal learning goal as well as helps him or her to work collaboratively to achieve the learning goal of his or her group. Since each intelligent agent is assigned to one specific user, it can adapt to his or her behavior and provide help and guidance to address his or her need. Furthermore, since the

intelligent agent resides on the computer of the learner, it does not overload any single computer or resource.

To address Principle 2, we utilize machine learning and multiagent learning techniques in our framework. According to (Mitchell 1997), *machine learning is the study of computer algorithms that improve automatically through experience*. Multiagent learning on the other hand is the learning process of the individual intelligent agents in a multiagent system from their own individual experiences (*centralized*) and from their interaction and communication with other agents (*decentralized*) (Sen et al. 2000).

As for Principle 3, we use multiagent coalition formation approaches to form multi-user learning groups. One reason for choosing multi-user learning groups over two-user learning groups is that during CSCL sessions, users learn from other students and also learn by teaching others (Chang et al. 2005). However, not all learner groups work well (Chalmers et al. 2005, Johnson et al. 1999). For example, in a two-user peer help scenario, the collaboration may suffer if one of those two participants is not happy working with the other participant. On the other hand, in a multi-learner group, the learners have a better chance of finding at least one member who they are happy to work with. Finally, we have chosen a multiagent coalition formation technique to form learner groups for the CSCL learning sessions. Each learner in our framework is represented and assisted by an intelligent agent. Automating multiagent coalition formation is possible since those agents, with minimal input from the learners, would be able to decide the best group for their learners to join because of their monitoring and modeling of the environment and their users. This would remove the burden of forming groups from the learners and the instructor.

I-MINDS Architecture

I-MINDS is based on a multiagent infrastructure in which the agents are intelligent—capable of flexible behavior and machine learning. These intelligent agents are autonomous and can operate robustly in rapidly changing, unpredictable, or open environments. With these intelligent agents serving and catering to students' unique needs and behaviors, students will be able to participate in a CSCL environment actively rather than listening to the lectures passively as in a traditional large or distance classroom. Currently, I-MINDS has three types of intelligent agents: (1) teacher agents, (2) student agents, and (3) group agents. A teacher agent, interacting with a teacher, is responsible for disseminating information streams to student agents, maintaining profiles for all students, assessing the progress and participation of different students, ranking and filtering the questions asked by the students, and managing the progress of a classroom session. In other words, the teacher agent provides support for the instructor to carry out the CSCL session. A student agent, on the other hand, mainly works as a personal helper to the student. The student agent manages the communication channels among students and between the teacher and the students. The student agent also presents the learning material to the student and forms groups with the other students for collaborative learning. The group agent is designed to provide support for the collaborative learning groups. The group agent monitors the students' interactions to evaluate each student's contribution as a group member.

The teacher agent, the student agents, and the group agents are designed to support the instructor, the students, and the groups, respectively. However, these agents need to communicate with each other to provide that support. There are two general types of messages that are exchanged among the agents: *directive* and *informative*. The *directive* messages are messages that are intended to control or modify the activities of the agents. For example, the teacher agent may request the student agent to

join a particular group using a directive message. On the other hand, an *informative* message is exchanged among the agents to transfer information. For example, the student agent may create a profile of its student user and send that profile to the teacher agent to be reviewed by the instructor. Table 1 shows the various messages that are exchanged among I-MINDS agents to provide CSCL support to the instructor and the students. Figure 1 shows an example of a CSCL session where the I-MINDS agents are communicating with each other.

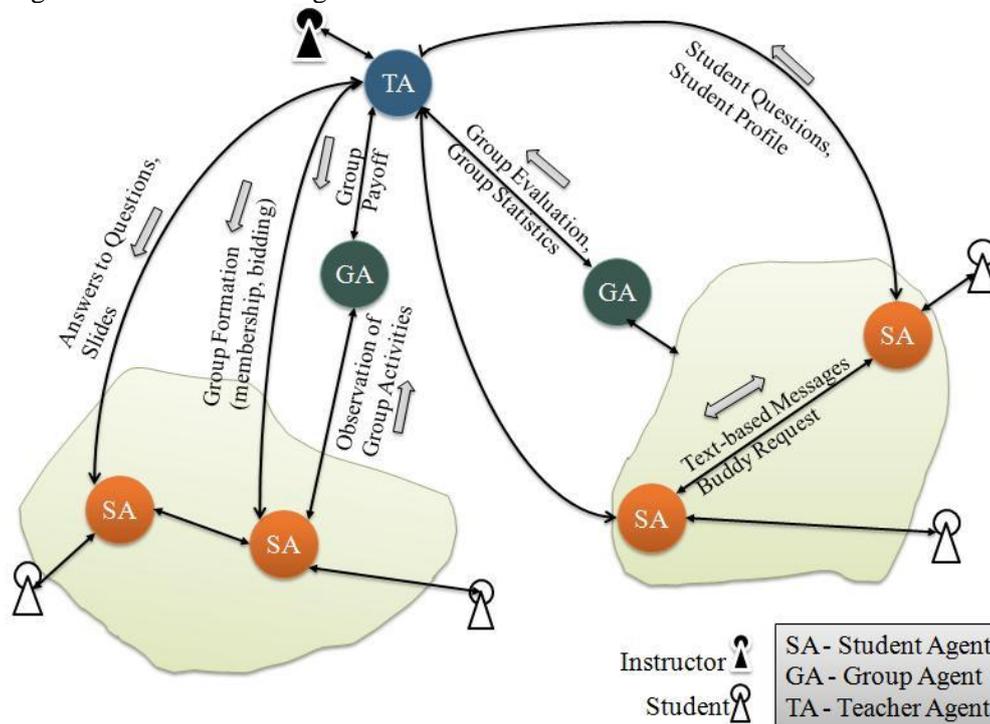


Fig. 1. Interactions and information exchanges among the agents of I-MINDS in a CSCL session.

In accordance to Principle 2 discussed earlier, our I-MINDS agents have machine learning capabilities. This is manifested in three important activities based on agent-agent interactions and agent-user interactions. First, the question ranking module of a teacher agent performs reinforcement learning—it learns new keywords and how to reweight keywords by observing how the instructor responds to a student question (in Equations (1)-(3) later). This allows the module to improve its ranking accuracy. Second, each student agent performs reinforcement learning—it learns how to estimate its student user's competence for a given task by storing the instructor's evaluation of the student user's performance in earlier tasks. This allows the student agent to put its student user in a group that is more appropriate based on the student's competence level (in the VALCAM-U algorithm later). Third, a student agent triggers multiagent learning. When calculating the compatibility between its student user and another student S , it combines the viewpoints or experiences from other agents' of that particular student S . This allows the student agent to put its student user in a more compatible group of users in future collaborative sessions (in the VALCAM-S algorithm later). Details of the machine learning mechanisms will be discussed in the following subsections on the three types of agents.

Table 1
Inter-Agent Communication for CSCL Support (SA = Student Agent, TA = Teacher Agent, GA = Group Agent)

CSCL Support	Message	Type	Sender	Receiver
Communication (SA)	Text-based messages	Informative	SA	SA
	Answers to questions	Informative	TA	SA
	Questions for teacher	Directive	SA	TA
Communication (TA)	Slides	Informative	TA	SA
	Answers to questions	Informative	TA	SA
Quiz Administration (TA)	Quizzes	Informative	TA	SA
Question Ranking (TA)	Student Profile	Informative	SA	TA
Buddy Group Collaboration (SA)	Student Profile	Informative	SA	SA
	Buddy request (join or removal)	Directive	SA	SA
	Buddy request response	Directive	SA	SA
Structured Group Collaboration Monitoring (GA)	Group evaluation	Informative	GA	TA
	Group statistics	Informative	GA	TA
Structured Group Performance Evaluation (TA)	Group payoff	Informative & Directive	TA	GA
Structured Group Formation (TA)	Group formation (bid values)	Directive & Informative	SA	TA
	Group formation (start bidding)	Directive	TA	SA
	Group membership	Informative	TA	SA
	Group formation (assignment of groups)	Directive	TA	GA
Structured Group Learning (TA)	Jigsaw cooperative learning sessions membership	Directive & Informative	TA	SA

Teacher Agents

In I-MINDS, the teacher agent is designed to support the instructor to carry out the CSCL sessions. The teacher agent allows the instructor to interact with students, send slides, manage Q&A sessions, administer quizzes, post evaluations, form groups and monitor individual and group performances.

Figure 2 shows the modules for an I-MINDS teacher agent. It has a GUI frontend, a database and file system backend, and a reasoning component with content-dependent and –independent modules. The content-dependent modules consist of knowledge or database specific to the subject matter or topics of a particular course, while the content-independent modules are generic features that add to the intelligence and capabilities of the agent.

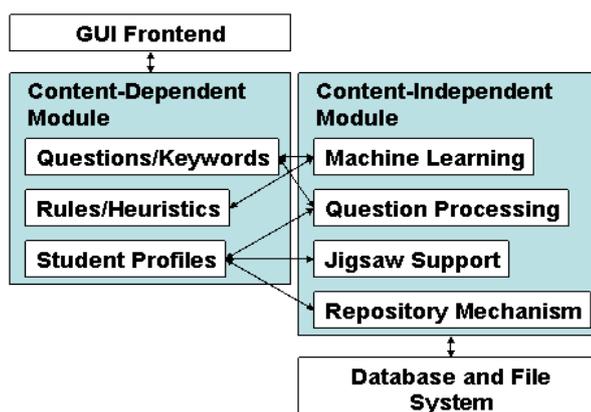


Fig. 2. Conceptual modules of an I-MINDS teacher agent.

Question ranking (Namala 2004) is a key capability of the Question Processing module shown in Figure 2 that allows the teacher agent to manage large or distance classrooms for the instructor. When questions are asked, the teacher agent scores each question based on a set of keywords and heuristics. Keywords are topic-specific specific and weighted while heuristics are course-specific. An example of a heuristic is: A question from a student who has been asking good questions will be ranked higher than a question from a student who has been asking poor questions, given that the two questions are exactly the same. A good question is also based on the number of weighted keywords that it contains and whether it is picked by the instructor to answer in real time. There are also exceptional heuristics being developed to account for scenarios such as “if a student has never asked a question before, then score his/her question high” to encourage their participation.

After scoring, the questions are ranked and displayed to the instructor. The instructor may choose to answer or discard a question, thus triggering the *reinforcement learning* mechanism of the module. Here is how a teacher agent’s reinforcement learning works. When the instructor answers or discards a question, the heuristic rules and keywords that were used to rank that question are updated using the following equation:

$$weight_{new} = weight_{old} + \Delta \quad (1)$$

To determine the value of the increment, Δ , if the question is answered,

$$\Delta = \left[\frac{Q_{score}}{Q_{max\ Score}} - \frac{Q_N - Q_{rank}}{Q_N} \right] \quad (2)$$

and if the question is discarded,

$$\Delta = - \left[\frac{Q_{score}}{Q_{max\ Score}} + \frac{Q_N - Q_{rank}}{Q_N} \right] \quad (3)$$

where, Q_{score} is the score of the question, Q_N is the total number of questions in the classroom, $Q_{max\ Score}$ is the maximum score of all the questions and Q_{rank} is the rank of the question. The logic behind the above increments is that if the teacher agent ranks a question high and it is discarded, then the heuristics and keywords that contributed positively to that high ranking should be penalized, while those that contributed negatively should be rewarded; and vice versa. Thus, by monitoring the

instructor's response to the questions received, the teacher agent learns how to better score and rank similar questions in the future simply by adjusting the weights of the keywords and heuristics in its database.

Note that the teacher agent supports the teacher in selecting the best questions to respond to. As described above, it computes a quality score for each question, ranks all questions, and presents them for the teacher to select. Thus, the decision on which questions are the best or appropriate to answer at any given time is ultimately made by the teacher. Thus, an individual teacher's point of view is always represented. Further, since the teacher agent is capable of reinforcement learning, it captures the decision (whether to answer or discard a question) and reviews its scoring strategies to increase the chance of accurately ranking future questions, in an attempt to learn each particular teacher's point of view over time. We see that as more heuristics are incorporated as I-MINDS evolves, the teacher agent will be able to not only learn, from a teacher, what keywords are important, but also which heuristics are applicable for which courses.

The teacher agent also helps the instructor by grouping similar questions together using the Question Group module. By grouping similar questions together, the teacher agent allows the instructor to address multiple similar questions together and profile students who ask similar questions. Question grouping is based on question classification and keyword matching. To achieve question classification, we adopt the *utterance classification* approach in the AutoTutor (Graesser et al., 1999). We use the ApplePie parser (Sekine & Grishman 1995) and the utterance classifier program of AutoTutor (Olney et al., 1995) to classify a question into one of twelve classes such as Contribution, Discovery, and so on. When a student asks the teacher a question using I-MINDS, the question processing module extracts keywords from that question. Then the similarity between two questions i and j is calculated using the following equation:

$$Similarity = 1 + \frac{n_{ij}}{n_i + n_j} + \frac{\sum_{l=1}^{n_{ij}} w_l^{ij}}{\sum_{l=1}^{n_i} w_l^i + \sum_{l=1}^{n_j} w_l^j} \quad (4)$$

Here, n_{ij} is the number of keywords that are common to i and j , n_i is the number of keywords in question i , n_j is the number of keywords in question j , w_l^{ij} are the weights of keywords common in both i and j , w_l^i and w_l^j are weights of keywords in i and j respectively.

Student Agents

In an I-MINDS supported classroom, a student agent serves a unique student. It interacts with the student and exchanges information with the teacher agent and the group agents. It also maintains a dynamic profile of its student user and a dynamic profile of the peers that the student has interacted with through I-MINDS, and forms a buddy group for its student user.

Figure 3 shows the conceptual modules of an I-MINDS student agent. Similar to the I-MINDS teacher agent, the design has a collaboration component instead of a question processing one.

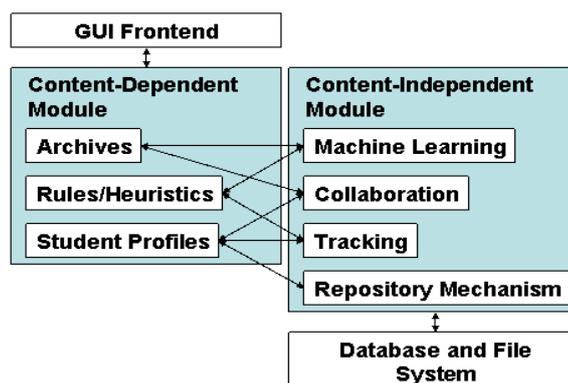


Fig. 3. Conceptual modules of an I-MINDS student agent.

Within the content-independent module, the tracking mechanism tracks the activities and the progress of the student. For example, I-MINDS provides a forum for class participants to communicate and view class participants' posts in the forum. These posts as well as the class participants' collaboration activities with other class participants are tracked and analyzed by a student agent. Other aspects of tracked behaviors include the number of messages each student contributed to the forum, the average length of each message, and the average quality of each message. I-MINDS also provides a digital whiteboard for class participants to have a visual discussion on a shared view. Aspects of tracked behaviors include the time that each student spent on the whiteboard and the tools—e.g., annotation, drawing, and eraser—that they used on the digital whiteboard.

The buddy group in I-MINDS is designed based on the model presented in (Hoppe 1995) that initiates and intelligently informs or parameterizes human-human cooperation. The model is based on what a student is supposed to know or is able to perform. For example, if a student is facing difficulty solving the assigned problem, he or she may ask for help. When such a request arrives, the system evaluates the cooperation criterion and prepares a list of potential helpers for him or her. Then a potential helper is asked whether he or she is willing to help. If the helper is willing, the asker and helper begin a cooperative session. If the helper declines, he or she is removed from the list of the potential helpers. In our implementation of buddy groups in I-MINDS, the student agent observes the questions posed by the student agent to the instructor. If the student is asking a lot of questions on a particular topic, it realizes that the student is facing difficulty understanding some concept. Then it automatically asks another student who had previously asked similar questions to act as a helper. The first student, who agrees to help becomes the buddy of the student who is having trouble understanding something. The students in the buddy group can communicate with each other using the forum and the token-controlled whiteboard in I-MINDS. After the buddy group is formed, the student agents keep track of their collaboration by monitoring their messages. If a student is not responsive, he or she is dropped from the buddy group.

A student is profiled by I-MINDS' agents in several ways. First, student-instructor interactions are profiled mainly based on the quality of questions that a student asks. Based on the grouping of these questions, students can also be profiled relative to others in terms of compatibility. Second, student-student interactions are profiled through the monitoring of a buddy group. This profiling process is carried out by a student agent from the viewpoint of the student it serves. That is, the student agent keeps a profile of its student's buddies in terms of their responsiveness in collaborative activities (forums and digital whiteboard). These include the number and frequency of messages sent,

type of messages, time spent on “idea sketching”, the number of times a student is dropped from a buddy group, and so on. Third, group interactions are monitored by the group agents during structured cooperative learning, which is detailed next.

Student agents are capable of reinforcement and multiagent learning as well, to be discussed when the VALCAM algorithm is presented later.

Group Agents

In I-MINDS, a group agent is activated when there are *structured* cooperative learning activities. When the students interact within their “buddy groups”, there are no group agents since the interactions are free-formed and not structured. However, each student is supported by its student agent. On the other hand, structured cooperative learning involves specified activities that explicitly require students to cooperate and thus group agents can be *more effective* since there is a structure to follow. Structured cooperative learning models explicitly specify how group activities are to be carried out in a sequence of steps to solve a joint task. Because of the steps, instrumentation points can be inserted into the process, which in turn allows the group agent to monitor and evaluate the groups more accurately. Activities instrumented or tracked include the number and type of messages sent among group members for each step, self-reported teamwork capabilities, peer-based evaluations as a team member, and evaluation of each team. Figure 4 shows the conceptual modules of an I-MINDS group agent. Note that this agent works entirely behind-the-scenes and thus does not have a GUI frontend. Included as part of its content-independent module are surveys and coalition formation features.

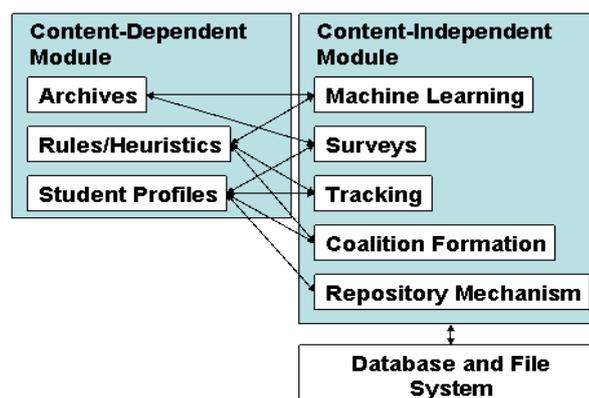


Fig. 4. Conceptual modules of an I-MINDS group agent.

To support structured cooperative learning, we have designed an *auction-based learning-enabled* coalition formation¹ algorithm called VALCAM (Soh et al., 2006b) that combines automatically tracked collaboration activities with subjective peer-based evaluations of a student’s performance in group work. However, not all student teams work: in some cases the students work in teams rather

¹ In this paper, we use “coalition” when the context is about multiagent decision making and forming teams of agents; and we use “group” when the context is about student peer learning and forming teams of students.

than working as a team (Chalmers et al. 2005, Johnson et al. 1999, Mulder et al. 2002). So, the learning goal of VALCAM is to, over time, improve the *competence* and the *compatibility* of each team. Here, the competence determines a learner's expected capability of solving a problem and the compatibility determines how well a group of learners are expected to work together. This approach to group formation is similar to the opportunistic group formation by (Ikeda et al. 1997) though the coordination protocol in (Ikeda et al. 1997) is through negotiation instead of auction, and the learner's roles are used to form groups instead of compatibility and competences.

Briefly, the VALCAM algorithm works as follows. Initially, the students or users are given some default amount of virtual currency to start with. The virtual currency is to be used as rewards to students who have performed well in the instructional activities, and to allow their student agents to bid to join more compatible groups in later rounds of activities. A typical coalition formation round starts with the instructor's command to the teacher agent. The teacher agent, acting as the auctioneer, announces a task, and then the student agents, acting on behalf of their corresponding student users, post their self-efficacy to a software blackboard architecture. Given a specific bidding protocol (Table 2), the teacher agent initiates a number of groups. Then, the teacher agent selects the first batch of student agents (which represent individual students) to join the groups following a specific group seed selection policy (Table 3). The teacher agent derives its perception of individual students' compatibility and competences based on the information it receives from the group and student agents. After the initial group(s) has been formed with their first members, the teacher agent announces the first round of bidding. The remaining student agents bid to join the most compatible and competent group depending on their respective virtual currency amounts. The teacher agent then selects the highest-bidders to join their respective groups. And then the teacher agent proceeds to announce the second round of bidding, and the process repeats until the last agent has been assigned to a group. With this setup, a student who has performed well individually (competence) and collaboratively (compatibility) will receive more virtual currency than one who has performed poorly. This in turn increases the likelihood that the student will be able to initiate or join a group that is more competent and compatible. Subsequently, the student is more likely to succeed in the subsequent collaborative sessions. On the other hand, a student who has not performed well will have a chance to join different groups for different sessions because of low bids, allowing him or her to explore the students who are not his or her first choices as group members, with the assumption that eventually a student will be able to identify a compatible group of students to work with. Once the groups are formed, the students interact with one another and solve the assigned task by communicating with one another with the help of the student agents. While students are working on the assigned problem, their group work is monitored and evaluated by the group agents. Finally, when the assigned task is completed, each student agent is rewarded with virtual currency based on its (student's) individual performance and its (student's) performance as a group member. Students who performed well will receive more virtual currency, allowing them to more successfully bid for their favorite groups in future sessions.

Here we present the detailed algorithm of VALCAM. Suppose that A is the set of *student agents*, m is the number of non-overlapping coalitions that will be formed, and $|A| > m$, $j \in J$ is the current task assigned, p is the selected auction protocol (e.g., English, Vickrey, etc. (Sandholm 1999)), b is the bidding protocol (Table 2 shows two example protocols), d is the group seed selection policy (Table 3 shows three example policies), and r is the default amount of virtual currency for new users. We also assume that each student is to be assigned to one and only one coalition. The VALCAM algorithm has two parts, VALCAM-S is for the teacher agent, VALCAM-U is for the student agents. That is, the VALCAM-U algorithm is executed by each student agent for its student user.

Table 2
Bidding Protocols

Protocol	Description	Characteristics
All Groups	Two or more groups are formed at the same time cycling through each group for bidding	Each group contains at least one competent user; find the best matching members for the members already in the group, and allow students to have a higher chance to join different groups; and allow “low-performing” students to have a higher chance to work with “high-performing” students
One Group	One group is formed at a time:	Best-matching students following the group seed selection policy will form the first group; and then the next best-matching students will form the second group; and so on, and allow students to have a higher chance to remain in the same groups, and allow similar-performing students to have a higher chance to work together

Table 3
Group Seed Selection Policies

Policy	Description	Characteristics
Competent User First	Select the most competent user to initialize a group	Each group contains at least one top competent user
Compatible User First	Select the most compatible user to initialize a group	Each group contains at least one top compatible user
Random	Select a user randomly from the to initialize a group	It is possible to have a group with non-compatible members or low-competence members.

VALCAM-S (A, m, j, b, p, r, d)

- 1) **Initialize:** i) Allocate r to the new student agents. ii) $L \leftarrow \lfloor |A|/m \rfloor$ iii) $G_f \leftarrow \phi$ iv) Announce task j to A
- 2) Select a set of *student agents* $q \in A$ and $|q| = m$, according to d , to form a set of new distinct user groups $G_f \leftarrow \{G_1, G_2, \dots, G_m\}$ where $|G_f| = m$ and $q_i \in G_i$
- 3) **While** $|A| > 0$ **do**,
 - i) **If** $|A| < L$, **then**,
 - a) Select random student agent $H \in A$ and random group $G_i \in G_f$
 - b) Collect B_{avg} from H . Where, B_{avg} is the average winning bid submitted by student agents in S
 - c) Assign H to S
 - ii) **Else For each** group $G_i \in G_f$ **do**,

- a) **If** $b = 1$ **then** $l \leftarrow L$
b) **Else** $l \leftarrow 1$
c) **While** $l > 0$, **do**,
 1) Start Auction protocol p for *student agents* in A
 2) Choose the highest bidder H
 3) Collect bid amount based on p from H
 4) $G_i \leftarrow G_i \cup H, A \leftarrow A - H, l \leftarrow l - 1$
4) Assign a group agent to each of the groups in G_f
5) **When** the assigned task j is completed, **do**, (i) Direct each *group agent* in G_f to distribute group payoff $V_g(s, j, t)$ to its student agent members based on their performance as team members, and (ii) Reward each *student agent* with individual payoff $V_s(s, j, t)$ based on its individual performance. Then the total payoff for a *student agent* s for task j at time t is defined as:

$$V_t(s, j, t) = w_{vg} \cdot V_g(s, j, t) + w_{vs} \cdot V_s(s, j, t) \quad (5)$$

where, w_{vs} and w_{vg} are the weights associated with the individual payoff and group payoff respectively, and $V_g(s, j, t)$ is the total group payoff for student agent s at time t for task j :

$$V_g(s, j, t) = w_{vpc} \cdot V_{pc}(s, j, t) + w_{vac} \cdot V_{ac}(s, j, t) \quad (6)$$

where $V_{pc}(s, j, t)$ and $V_{ac}(s, j, t)$ are the payoffs distributed by the group agent to the student agent based on the potential and actual contribution of the student agent as a member of the group, respectively; w_{vpc} and w_{vac} are the weights associated with contributions. Furthermore,

$$V_{pc}(s, j, t) = \frac{PC(s, j, t)}{\sum_{i \in G} PC(i, j, t)} \quad (7)$$

where $PC(s, j, t)$ is the potential contribution of student agent s at time t for task j and G is the set of all student agents in the group of student s . The potential contribution is an expected measure of the student's performance for the assigned task. Moreover,

$$V_{ac}(s, j, t) = \frac{AC(s, j, t)}{\sum_{i \in G} AC(i, j, t)} \quad (8)$$

where $AC(s, j, t)$ is the actual contribution of student agent s at time t for task j and G is the set of all student agents in the group of student s . If a student agent performs well as a group member (e.g., responsive and helpful towards other agents and contributes to team goals.) then this value will be high. This global condition is set to motivate the agents to be a team player.

We also measure the compatibility between two agents s_1 and s_2 , from the viewpoint of s_1 , as:

$$C_p(s_1, s_2, t) = w_m \cdot E_m(s_1, s_2, t) + w'_m \cdot E'_m(s_1, s_2, t) \quad (9)$$

where $E_m(s_1, s_2, t)$ denotes a numeric value computed by s_1 representing its experience in working with s_2 at time t , $E'_m(s_1, s_2, t)$ denotes the evaluation score that s_1 receives from s_2 , w_m and w'_m are

weights. We further compute the time averaged compatibility measure between two agents s_1 and s_2 as:

$$\bar{C}_p(s_1, s_2, t) = \frac{\sum_{t=0}^{q-1} C_p(s, j, t)}{T} \quad (10)$$

In Eq. 10, t denotes a time period, T is the number of time periods between $t = 0$ and $t = q - 1$, w_t is the weight associated with the time period t .

Note that VALCAM-S is an algorithm that uses multiagent learning to improve the quality of the groups formed over time. The experience gained from each iteration of group formation is distributed and stored at each student agent. During the bidding process, each student agent computes the compatibility of its student user with all student members of each group that it is considering to join. This computation requires the student agent to combine the experiences of all student members of a group with its own student user, and in the end, selects the group with the best overall compatibility (as well as competence) to join. This search process is gradually improved with updated experiences that are reinforced by the subsequent interactions among group members and captured by their respective student agents. This basically allows the overall system to conduct a “guided” trial-and-error search and learn how to form the best student groups.

VALCAM-U (A, m, j, b, p, r, d)

- 1) **If** selected as one of the group seed, **then** Exit
- 2) Post the competence measure c_m to the common bulletin board after a new task announcement. c_m is basically the measure of the *student agent's* ability to perform the announced task. The competence measure of the student agent s for task j at time t is defined as:

$$C_m(s, j, t) = \frac{w_{ce} \cdot U_e(s, j, t) + w_{ck} \cdot U_k(s, j, t)}{w_{ce} + w_{ck}} \quad (11)$$

where, $U_e(s, j, t)$ is the mean of previous evaluations received by the student agent s on tasks similar to j at time t , $U_k(s, j, t)$ is the student agent s ' knowledge or expertise on the assigned task j at time t , w_{ce} and w_{ck} are weights.

Notice that the student agent uses *reinforcement learning* (Equation 11) to calculate the competence of its student user. Every time a student is evaluated on his or her performance on a particular task, the student agent stores that evaluation score along with the task description—capturing how well the student user performed that task. This stored experience is then used to compute $U_e(s, j, t)$, which in turn improves the accuracy of the competence measure.

- 3) **When** acceptance of bid for the i th group is announced by the *teacher agent*, repeatedly bid an amount of virtual currency $V_b(s, j, t)$ until a bid is won. Here,

$$V_b(s, j, t) \propto \frac{\sum_{k=1}^y [C_m(k, j, t) + \bar{C}_p(s, k, t)]}{Y} \quad (12)$$

where, $V_b(s, j, t)$ is the virtual currency bid submitted by the student agent s at time t for task j , $C_m(k, j, t)$ is the competence measure of the k th member of the i th coalition for task j , $\bar{C}_p(s, k, t)$ is the time-averaged compatibility measure between student s and student k at time t and $k \in A$.

For our current implementation, VALCAM uses the “All Groups” bidding policy and a hybrid of the “Competent User First” and “Compatible User First” group seed selection policies. That is, VALCAM selects users who have been competent and worked well with other users as first members of the coalitions.

Jigsaw Learning Model

The Jigsaw cooperative learning model was first introduced by (Aronson et al. 1978). This procedure works as follows. First, the instructor partitions the students into main groups. Second, the instructor divides a problem into different parts (or sections). Third, the instructor assigns a part/section of the problem for every student such that members of the same *main* group will have different parts/sections to solve. The students who are responsible for the same section then work together in a *focus* group to come up with solutions to the section to which they have been assigned and develop a strategy for teaching the solutions to their respective (main) group members. Clarke (1994) further refined the Jigsaw structure into four stages. These stages are (1) *Introduction* of the topic to the class as a whole, (2) *Focused Exploration*: The focus groups explore issues pertinent to the section that they have been assigned, (3) *Reporting and Reshaping*: The students return to their original main groups and instruct their teammates based on their findings from the focus groups, and (4) *Integration and Evaluation*: The team (main group) connects the various pieces generated by the individual members, addresses new problems posed by the instructor, or evaluates the group product.

An Example of Use

The instructor and the students together with the I-MINDS agents carry out the I-MINDS CSCL session. In the beginning of the session, the instructor opens a new session with the teacher agent and registers the keywords (and their weights) and heuristic rules for question ranking module. The instructor may opt to bypass this step since the teacher agent is capable of learning new keywords and their weights as it interacts with the instructor during the course of the session. When a student logs into the virtual classroom, he or she is assigned a student agent. During the session, the instructor may start by using slides, or administering a quiz to the students through I-MINDS. Each student agent archives the received instructional material for its student user. At the same time, a student user may communicate with the instructor in the form of text-based questions and communicate with his or her peers through the whiteboard or forum features of I-MINDS. When the teacher agent receives a question from a student, it proceeds to score, rank, classify, and group the question before presenting it to the instructor. The instructor may choose to answer or discard a question. By doing so, the instructor is providing positive or negative reinforcement to teacher agent’s question ranking strategies. Concurrently, a student agent keeps a profile of its student user; for example, monitoring how well his or her questions have been received by the instructor. Based on this profile, a student may be requested to join a buddy group; likewise, if a student has been found to be un-responsive to his or her peers’ forum or whiteboard discussions, the student may be requested to leave the buddy group. Further, the instructor can also request to have a structured group learning session in the classroom.

The instructor may do this through the Jigsaw cooperative learning module. This is where the VALCAM algorithm is used to form student coalitions to solve joint tasks in phases. A group agent will be spawned dynamically to oversee each group's activities, serving as a "mediator" between the teacher agent and the student agents in the group. Also, after the structured group activities are completed, the student agents work behind-the-scenes and exchange information to learn about their respective student users' competence and compatibility. At the end of the session, the instructor can view the collaborative performance reports of the students from the group agents. The instructor can then post individual evaluations of the students using the teacher agent. The student agents and the group agents receive these evaluations and store them to improve the compatibility and competence level of student structured groups formed in future sessions.

During a session, students may log out and log back in at a later time. The session ends when the instructor terminates it. All data will be stored through the repository mechanisms as shown in Figures 2-4 by the agents, for future use.

IMPLEMENTATION

I-MINDS is built using a loosely-coupled layered architecture (Figure 5), which makes I-MINDS flexible and usable in heterogeneous environments.

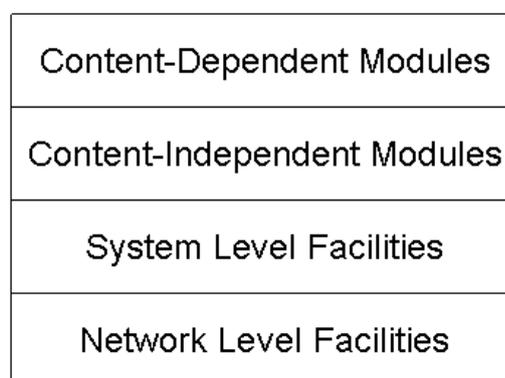


Fig. 5. I-MINDS architecture.

The bottom layer (i.e., the network layer) in I-MINDS provides the basic communication functionalities by using TCP/IP sockets. Since sockets are available in a variety of platforms across the network, this layer allows I-MINDS to communicate in heterogeneous situations.

The second layer provides system-level protocols and encapsulations with necessary abstractions to provide convenient communication and deployment functions to the upper layers. This level includes a Relational Database Management System (RDBMS)-type database (MySQL) and an audio/video server (Macromedia Flash[®] Communication Server). The database is used by the agents for fast storage and retrieval of information and the audio/video server helps establish audible and visual communication between the teacher and the students.

Finally the topmost two layers of I-MINDS contain the intelligent agents i.e., teacher agents, student agents, and group agents. Each intelligent agent has two sets of modules: content dependent and content independent. The content independent set provides the definitions and processes for general education-related services, while the content-dependent module handles specific course-

related information and knowledge base, providing the required data and the heuristics used to gather, analyze, disseminate and process the generated data.

Figure 6 shows an example of the topological infrastructure of I-MINDS. The manager manages the ongoing classroom sessions. It manages system level information such as the list of ongoing classes, list of courses and teachers, the login names and passwords for the students for each of the classrooms, etc. I-MINDS is capable of holding multiple concurrent classroom sessions. In each I-MINDS classroom session, the teacher, students and student groups are assigned the teacher agent, student agents and group agents, respectively. These agents support the person/group that they serve.

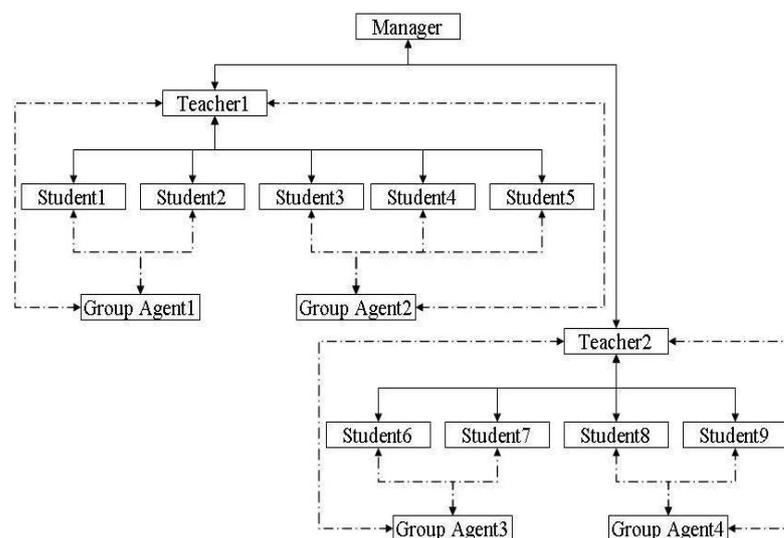


Fig. 6. I-MINDS agent topology

For our research prototype and evaluations, the I-MINDS system was implemented in Java (SDK 1.4.2). We have used Java's socket functionalities to establish communication among agents, Java's Swing class to create interfaces, and Java's JDBC technologies to connect to our MySQL database repositories to store and retrieve all data. Presently, we continue to develop our research prototype in Java. In parallel, we have also ported most of the I-MINDS features to Microsoft's Conference XP platform where the audio/video streaming, networking, archiving, tracking, and communication infrastructures are readily available. This porting has allowed us to deploy our system in wired and wireless environments and with more robust communication modes and data storage.

Currently, I-MINDS also has the following features: superimposition of handwriting on lecture screen, superimposition of text on archived lecture notes, archival, multicast/broadcast, multi-character forum (colors and fonts), multi-character digital whiteboard (colors, fonts, and exclusive tokens), capture of desktop as slides, and annotation of archived lecture notes. The capture of desktop as slides allows a teacher to capture anything on display on his or her computer's desktop (e.g., a webpage, a document, a Powerpoint presentation, or a directory) as a slide, annotate the slide and broadcast the slide to the students.

Teacher Agent

Figure 7 shows a screen snapshot of our teacher agent interface. The snapshot shows three components. First, the main window displays the lecture materials which could be a PowerPoint slide as shown, a whiteboard (captured with a Mimios-based technology), a Web page, and/or any documents that appear on the computer screen. In Figure 7, the lecture material happens to be a Microsoft PowerPoint slide on buffer zones, a topic in Geographic Information Systems (GIS). Second, the figure has a small toolbar, shown here at the top-left corner of the snapshot. Only an instructor can view and use this toolbar. This toolbar allows the instructor to save and/or transmit a learning material and change the annotation tools (pens, erasers, markers, and colors). Third, the snapshot shows a question display window at the bottom right corner. Once again, only the instructor can view and use this question display. The question display summarizes each question, ranked based on their scores. The display window also has several features. For example, an instructor may choose to answer or discard a question, may view the entire question, and may review the profile of the student who asked a particular question. Alternatively, the instructor may choose to send the toolbar and the question display window to the background so as not to interfere with her or his lecture materials.

The screenshot shows a presentation slide titled "Buffer zones: buffer zones" with a blue polygon buffer. The slide content includes:

- Region within 'x' distance units
- Buffer any object: point, line or polygon
- Use multiple buffers at progressively greater distances to show gradation
- May define a 'friction' or 'cost' layer so that spread is not linear with distance

Examples:

- 200 foot buffer around property where zoning change requested
- 100 ft buffer from stream center line limiting development
- 3 mile zone beyond city

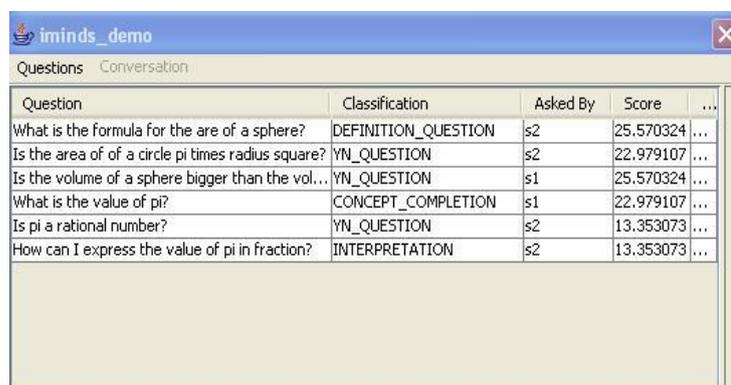
Diagrams show "point buffers" (three circles) and a "line buffer" (a thick line). A toolbar at the top left includes "Save/Return", "Pen", "Eraser", "Marker", and "Color". A question display window at the bottom right shows a table of questions:

Questions	Conversation	Statistics	
Question	Asked By	Timestamp	Score
we cant hea...	is12	Jun 4, 2003	0
sound is out...	is12	Jun 4, 2003	0
is12	is6	Jun 4, 2003	0
hi	is9	Jun 10, 2003	0
what is GIS?	is8	Jun 10, 2003	13

Fig. 7. Snapshot of an I-MINDS teacher agent delivering a lecture note annotated at real-time.

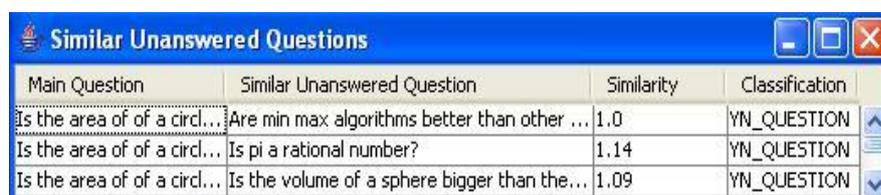
Figure 8(a) shows the popped up Q&A interface for the teacher agent. Each question is displayed with its type classification, the student who asked the question, and a score. The instructor then could sort the questions by clicking at a specific column heading. In Figure 8(a), the questions are sorted by their "scores", where the scores are computed using the design as discussed earlier. The instructor may also click on each question to bring up a detailed information window that displays the entire question and the profile of the student who asked the question. The instructor may also click on each question to "discard" or "answer" it, effectively "teaching" the teacher agent to learn about the inappropriate or appropriate questions. The Q&A interface also has a menu group "Questions" that allows the instructor to bring up "similar unanswered questions" of a question. Figure 8(b) shows the

graphical user interface of this component. It lists the main question, the similar unanswered questions, the similarity scores, and the type classification.



Question	Classification	Asked By	Score	...
What is the formula for the are of a sphere?	DEFINITION_QUESTION	s2	25.570324	...
Is the area of of a circle pi times radius square?	YN_QUESTION	s2	22.979107	...
Is the volume of a sphere bigger than the vol...	YN_QUESTION	s1	25.570324	...
What is the value of pi?	CONCEPT_COMPLETION	s1	22.979107	...
Is pi a rational number?	YN_QUESTION	s2	13.353073	...
How can I express the value of pi in fraction?	INTERPRETATION	s2	13.353073	...

(a)



Main Question	Similar Unanswered Question	Similarity	Classification
Is the area of of a circl...	Are min max algorithms better than other ...	1.0	YN_QUESTION
Is the area of of a circl...	Is pi a rational number?	1.14	YN_QUESTION
Is the area of of a circl...	Is the volume of a sphere bigger than the...	1.09	YN_QUESTION

(b)

Fig. 8. Snapshots of question processing features: (a) scoring and details of each question, (b) grouping of similar questions.

Figure 9 shows a screen snapshot of the I-MINDS student agent, which is divided into four major quadrants. The top-left quadrant displays in real-time the lecture materials delivered from the teacher agent to each student agent. When the instructor changes a page, for example, the teacher agent will send the new page to the student agent. The student agent duly displays it. Further, when the instructor writes on a page, the teacher agent also transmits the changes to the student agent to display them for the student. The top-right quadrant is broken up to two sub-regions. On the top is a real-time video feed from the teacher agent, which can be optionally hidden in the background. On the bottom is the digital archival repository of the lecture pages. A student may bring up and annotate any archived page. For example, he/she might paste a question onto a page and send it back to the instructor as a “question with a figure.” On the bottom-left quadrant is the forum. Each message posted is color coded and labeled with the ID of the student who posted the message. On the bottom-right quadrant is the set of controls for asking questions. A student can type in his or her questions here, and then send the questions to the instructor, to the buddy group, to both the instructor and buddy group, or to a particular student in the buddy group. A student can also invite other students to join his or her buddy group through the *invite* function found in this quadrant.

The student agent interface has a menu bar on top, with menus labeled *Class*, *Presentation*, *Forum*, *Slides*, *Collaboration*, and *Help*. The *Class* menu has features pertinent to registration, login, and setup of a class lecture. The *Presentation* menu contains options on the lecture pages such as sound, colors, annotations, etc. The *Forum* menu allows a student to setup and manage his or her

forums. The *Slides* menu allows a student to archive, search, retrieve, and generally manage all the archived lecture pages. Finally, the *Collaboration* menu provides options on features that support collaborative activities—grabbing a token of the digital whiteboard, initiating a digital whiteboard discussion, turning off the automated buddy group formation, and so on.

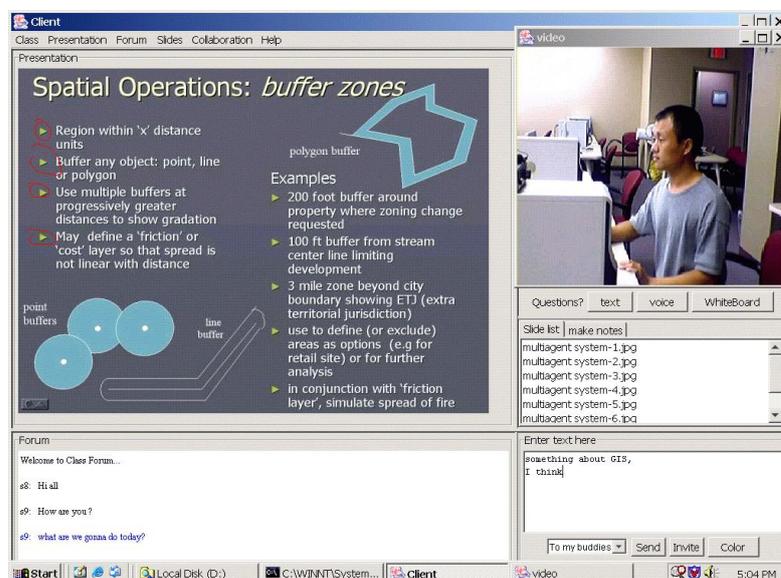


Fig. 9. Snapshot of an I-MINDS student agent receiving a lecture note, a video stream from *the instructor*, and its student using the forum.

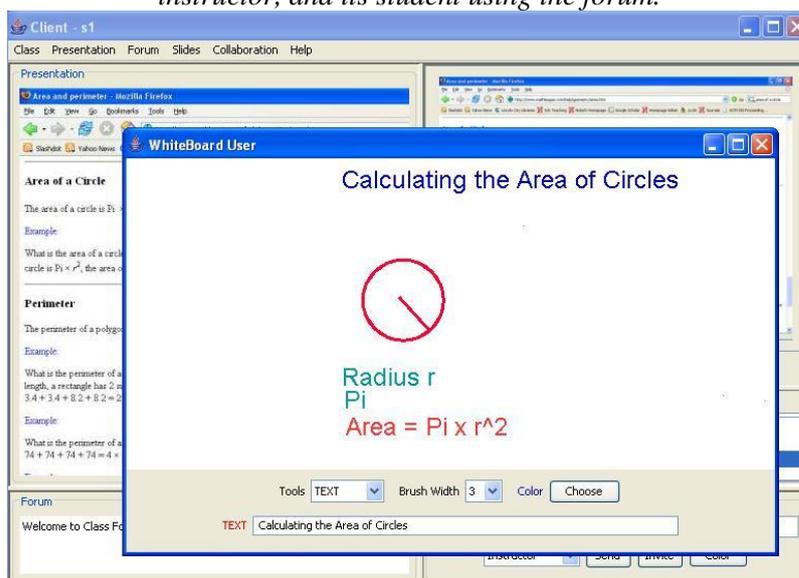


Fig. 10. Snapshot of a student using the digital whiteboard provided by its I-MINDS student agent.

Figure 10 shows the student agent interface with the invoked digital whiteboard. It allows students to draw and write on the board simultaneously through their student agents. Further, it has a

“control token” that a student may grab such that he or she can draw and write on the board while the others cannot. This allows a student to be an active contributor while others in the buddy group “watch” or “listen” to what the student is going to contribute. The student may then release the token to allow for group contribution to resume.

RESULTS AND DISCUSSIONS

Two-Semester Study: Collaboration and Structured Cooperative Learning

To study the impact of I-MINDS on structured cooperative learning, a two-semester study was carried out in the closed labs of CSCE 155 at the University of Nebraska, in the Spring and Fall semesters of 2005. CSCE 155 is the first core course of computer science and computer engineering majors (i.e., CS1). The course has three 1-hour weekly lectures and one 2-hour weekly laboratory sessions. In each lab session, students were given specific lab activities to experiment with Java and practice hands-on to solve programming problems. For each semester, there were 2-3 lab *sections* where each section had about 15-25 students.

The study utilized a control-treatment protocol. In the *control* section, students worked in Jigsaw cooperative learning groups without using I-MINDS. Students were allowed to move around in the room to join their Jigsaw groups to carry out face-to-face discussions. In the *treatment* section, students worked in Jigsaw cooperative learning groups using I-MINDS. Students were told to stay at their computers and were *only* allowed to communicate via I-MINDS. With this setup, we essentially simulated a distance classroom environment.

For each lab, the students were given a lab handout with a list of activities—thus, a lab is a task and its activities are the subtasks. We conducted the study for several lab sessions, covering topics in debugging and testing, inheritance and polymorphism, Unified Modeling Language (UML), and recursion.

The students of both the control and treatment sections were required to complete the tasks and subtasks in the four Jigsaw phases as discussed earlier.

In each section, the instructor announced the main groups. In the control section, this was done manually. In the treatment section, I-MINDS automatically performed group formation (coalition formation) using the VALCAM algorithm. Once the main groups were formed, the teacher agent formed the focus groups by randomly selecting students from the main group. After the focus groups were formed, every focus group was assigned one subtask randomly. After the subtask assignment, the focused exploration phase was started. Then the three remaining Jigsaw Phases were carried out in order. During these three phases, the student agents and the group agents monitored and guided the activities of the students and the student groups, respectively. After the three Jigsaw Phases were executed, all the students filled out the Peer Rating Questionnaire and Team-Based Efficacy Questionnaire and took a 10-minute post-test. This 10-minute post-test score was used as the measure of student performance in terms of understanding the topic of the lab.

First, we look at the average normalized post-test scores, as shown in Figures 11 and 12. Each normalized score is computed by dividing each student’s post-test score for a test day by the sum of the student’s post-test scores of all other lab days that did not involve the Jigsaw experiment. Therefore, this normalized score provides a measure to compare the performances of the control and treatment section students in a scale that does not depend on the individual student’s abilities. For both Fall 05 and Spring 05 experiments, a *t*-test reveals that the average normalized post-test score of

the treatment section is significantly higher than the average normalized score of the control section ($\alpha = 5\%$, p value for Spring is 6×10^{-4} and Fall is 5×10^{-5}). This indicates that the I-MINDS-enabled structured cooperative learning outperformed the conventional face-to-face one. Initially, we had expected that students in the control group to perform better than students in the I-MINDS group. This is because we realized that the I-MINDS implementation did not capture all interactive cues that traditional face-to-face interactions provide. For example, important cues such as facial expressions, intonations, hand gestures and eye movements that can be observed when students interact face-to-face are lost when students interact through I-MINDS as they cannot see each other. However, we also had an expectation that the current I-MINDS implementation could still outperform the traditional face-to-face setup because of the studies from (Picciano 2002) and (Beaudoin 2002) where students had to articulate more carefully due to the lack of other visual cues. This result indicates that the students in treatment section might have been compelled to articulate more carefully and clarify their ideas to convey them to their team members. And such articulation might have helped them achieve better scores than the control section students.

We also observe that students in the treatment sections seemed to improve overtime, and their performance seemed to eventually overtake that of the control sections' over time (see Figures 11 and 12)—indicating that VALCAM, due to its learning mechanism, might have been effective in forming better and better coalitions over time. However, these promising results will require more data collection in the near future to attain higher significance.

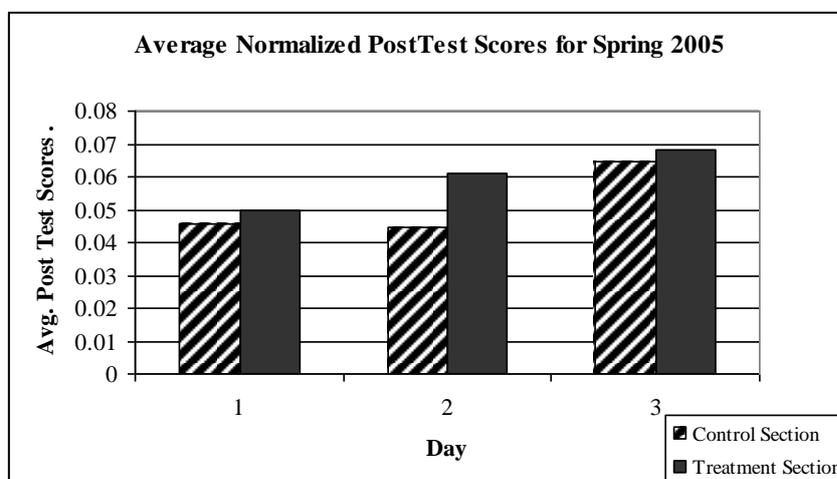


Fig. 11. *Control vs. Treatment*: Average normalized post-test scores for Fall 2005; each student's score for each day is divided by the student's average scores on all other lab days not involving the Jigsaw experiment.

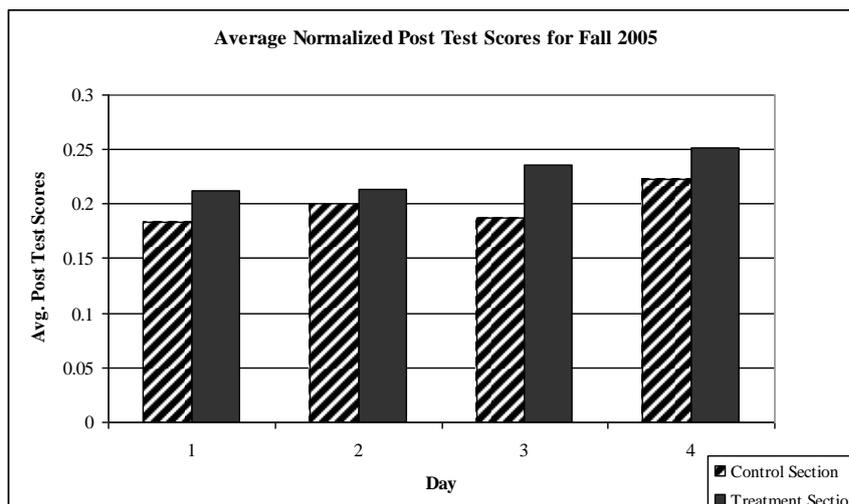


Fig. 12. *Control vs. Treatment*: Average normalized post-test scores for Fall 2005; each student's score for each day is divided by the student's average scores on all other lab days not involving the Jigsaw experiment.

Next, how students perceived the quality of their group activities is analyzed based on two surveys: the *Peer Rating Questionnaire* (PRQ) and the *Team-Based Efficacy Questionnaire* (TEQ).

The Peer Rating Questionnaire (PRQ) surveys were conducted in both control and treatment sections *after* each lab session was completed. The PRQ is designed to quantify the compatibility of the group members after they have gone through the cooperative learning process. The average peer rating scores that each student gave to his or her group members for each section can be used as a measurement of how well the team members in each section were able to work with each other. Table 4 shows the results of the PRQ surveys.

Table 4
Control vs. treatment sections: Results of the peer-rating questionnaires.

Fall 2005				
Session	Control Section		Treatment Section	
	Mean	Stdev.	Mean	Stdev.
1	42.10	2.73	32.45	5.78
2	36.62	7.05	37.72	4.60
3	39.91	4.80	34.63	8.08
Mean	39.54	4.86	34.93	6.15
Spring 2005				
Session	Control Section		Treatment Section	
	Mean	Stdev.	Mean	Stdev.
1	35.39	2.30	33.71	4.69
2	34.87	5.32	35.80	12.21
3	36.03	3.19	36.37	5.18
4	37.53	3.37	37.25	3.62
Mean	35.95	3.54	35.78	6.42

As evidenced in the scores, students in the control section rated their peers better (higher means) and more consistently (lower standard deviation values) than the students in the treatment section. This is possibly due to the face-to-face interaction. After all, students interacting through I-MINDS could not enjoy the advantages of face-to-face interactions such as facial expressions, the spontaneous free-flowing of ideas, and more immediate feedback in their discussions. This observation indicates that I-MINDS still lacks sufficient GUI features and multimedia capabilities to fully capture real-time characteristics of interactions.

On the other hand, there are indications that students in the treatment section for the Spring 2005 section seemed to rate their peers better over time (from 33.71 to 35.80 to 36.37 and 37.25). This might be due to the ability of the coalition formation algorithm in forming better groups over time. Given sufficient amount of time the VALCAM algorithm for coalition formation would allow the users to form groups with their favorite peers. So, over time the users were able to interact with others and evaluate each others as team members. This evaluation in the form of PRQ then helped them choose better team members in the future sessions. However, the high and fluctuating standard deviation indicates that we need to collect more experiment data to perform a conclusive comparison between the PRQ values of the control and treatment section.

The Team-Based Efficacy Questionnaire (TEQ) surveys were collected after each lab based on a set of questions designed to measure how a student viewed how well his/her group had performed, as shown in Table 5. It is observed that students in the control section approved of their team-based activities more than the students in the treatment section. There are two possible explanations. First, the ease of face-to-face interactions gave the impression that the group was doing better, which is consistent with our earlier observation with the peer rating results. Second, how the student agents form their coalitions did not necessarily meet the students' preference. Note that a student did not have access to other survey results, including how his or her group members thought of him or her as a peer. However, the student agent did and perused this information in its bidding for the most useful/compatible group. Further studies will be necessary to investigate how a student's perception of a group correlate with or influence the actual quality of a group.

Table 5
Control vs. Treatment sections: Results of the team-based efficacy questionnaires

Spring 2005				
Session	Control Section		Treatment Section	
	Mean	Stdev.	Mean	Stdev.
1	31.80	2.58	27.72	5.08
2	30.87	3.38	29.18	2.63
3	30.08	3.02	28.25	4.02
Mean	30.92	2.99	28.38	3.91
Fall 2005				
Session	Control Section		Treatment Section	
	Mean	Stdev.	Mean	Stdev.
1	27.22	4.37	23.64	5.55
2	26.75	6.66	25.87	8.33
3	29.14	5.47	25.76	5.43
4	29.12	4.52	26.78	8.15
Mean	28.05	5.25	25.51	6.86

Finally, the correlation between a student's performance and the other parameters is investigated, as shown in Table 6. First, it is observed that the treatment section had higher correlation values in SEQ (0.41 vs. 0.28), PRQ (0.34 vs. 0.23), and (0.50 vs. 0.22) than did the control section. This indicates that the better students (with higher post-test scores) in I-MINDS groups rated their self-efficacies better, rated their peers better, and rated their team-based efficacies better than those in the traditional face-to-face groups. Looking more closely at how I-MINDS students interacted, we see that students who had better post-test scores were also the students who sent longer messages (with a correlation of 0.40). Thus, in this case, better students assumed a larger role in their respective groups during the treatment. Combining this observation with what has been reported earlier on the average normalized post-test scores, there are indications that better students helped other students better in the treatment section and that resulted in better individual performances as evidenced in the post-test scores.

Table 6
Correlation between Post-test Score and Other Parameters

Spring 2005		
Variables	Control Section	Treatment Section
	Corr. w/ Posttest	Corr. w/ Posttest
SEQ	0.28	0.41
PRQ	0.23	0.34
TEQ	0.22	0.50
# of Messages Sent	N/A	0.11
Avg. Length of Messages	N/A	0.40
Fall 2005		
Variables	Control Section	Treatment Section
	Corr. w/ Posttest	Corr. w/ Posttest
SEQ	0.22	0.16
PRQ	-0.01	0.11
TEQ	0.09	0.12
# of Messages Sent	N/A	0.27
Avg. Length of Messages	N/A	0.25

However, the above observations were not repeated in the Fall 2005 study. Comparing the two sections, how students did during the cooperative learning activities did not correlate with how they performed individually in the post-tests. Compared to Spring 2005, better-performing students in Fall 2005 tended to send more messages (0.27 vs. 0.11), but shorter messages (0.25 vs. 0.40). Does that mean that the better-performing students in Fall 2005 were less patient with their peers? Further, students in the Spring 2005 treatment section reported a 0.41 correlation between their self-efficacies and their post-test scores, compared to only 0.16 in the Fall 2005 treatment section. That means the students in Fall 2005 treatment section were far less accurate in their knowledge of their own ability to solve the upcoming problem set. This knowledge is very important to form effective teams: students who think they are good at a particular topic when in fact they are not as good can misguide the group activities.

This indicates that even though a student is doing very well individually, he/she may not be helpful to other team members. Therefore, when forming coalition of learners, the individual competence is not the only factor, the compatibility and the history of the members working together should also be considered. Considering this factor, I-MINDS' coalition formation algorithm considers the PRQ values while forming teams. However, since this algorithm uses reinforcement learning, it needs some training before it could form effective cooperating teams. The quality of interaction between the teammates depends on various things, their likings of each other, their expertise on the problem and the difficulty of the assigned problem. This last factor is vital because if the problem is too easy, interaction between group members becomes a liability instead of being an asset. From our close observation of the students, it was observed that more students in Fall 2005, on average, found the assigned problems to be easy, than those in Spring 2005—they also achieved better course grades. Therefore, this could be a very possible reason for the lack of impact of structured cooperative learning (both control and treatment) in Fall 2005. This hints that a learner coalition in a CSCL environment could work better only when the problem is sufficiently difficult for one group member to solve by himself/herself. This could then also motivate students to exchange messages to help each other obtain a solution.

The above comparative and correlation analyses provide some valuable ideas to examine further:

- Artificially intelligent tools like I-MINDS can perform comparably with the traditional face-to-face interaction in cooperative learning setting, as indicated by the higher average normalized post-test scores, and PRQ and TEQ values of the treatment sections.
- In a structured CSCL environment, the participation of the better-performing students and thus the success of a coalition, somewhat depends on the difficulty of the problem assigned. The low correlation between TEQ, the number of messages exchanged, the average length of messages exchanged and the post-test scores indicate that. If the problem is too easy, the better-performing students may not be eager to work with the weaker students and that can reduce the quality of the teamwork.

Third, a better-performing student is not necessarily a more effective member of a cooperative learning group. The low correlation between PRQ, TEQ and the post-test scores of the students hint at this idea. This validates the I-MINDS approach in including peer-based ratings to determine the virtual currency rewarded to each student.

RELATED WORK

In this section, we focus our discussion of related work to agent-based or agent-flavored systems for supporting collaborative learning. Agent-based systems are such as KA-Agent (Ogata and Yano 2000), I-Help (Vassileva et al. 2002, Bull et al. 2001, Vassileva et al. 1999), MATHNET (Teixera et al. 2003), and COLER (Constantio-González et al. 2003). And agent-flavored systems are such as DEGREE (Barros and Verdejo 2000), MArCo (Tedesco 2003), and a simulated student (Vizcaíno 2005). The latter group of systems does not explicitly discuss agents or multiagent systems *per se* but their systems possess most of the characteristics shared by agents.

While discussing these CSCL systems, we focus mainly on the CSCL support they provide. There are mainly three categories of support provided in a CSCL environment: (1) *learner support* – designed to help the learner interact and learn from the instructor and peers, (2) *instructor support* – designed to facilitate the instructor's monitoring and supervision process of the CSCL environment,

and (3) *collaborative support* – designed to improve the collaboration among the students in the CSCL environment.

Ogata and Yano (2000) discussed how they use the knowledge awareness, i.e., informing a learner about other learner's knowledge, to enhance learner collaboration in Sharlok—an open-ended CSCL environment. The agents in Sharlok observe the actions of a learner and then notify him or her about other learners (1) who are having the same problem, (2) who have different point of view about the problem, and (3) who have the potential to solve the problem. In terms of learner support, both Sharlok and I-MINDS provide basic communication (e.g., chat, whiteboard, etc.). Moreover, Sharlok provides knowledge awareness to the learners by filtering user communication and by creating brief summaries of the other learner's activities. Unlike I-MINDS, Sharlok does not provide any instructor support (e.g., communication with users, monitoring and evaluating learners, etc.). Also, I-MINDS provides group formation, monitoring of group activities, and evaluation of group performances of the students as collaborative support. However, Sharlok does not provide any such collaborative support.

I-Help (Vassileva et al. 2002, Bull et al. 2001, Vassileva et al. 1999) is built on a multiagent architecture that combines a one-to-one network and a discussion forum to provide offline peer help to learners. Each learner in I-Help is represented by an agent who models his or her knowledge and behavior. Whenever a learner seeks help, his or her representative agent communicates with the other agents in the system and find the most suitable learner who could provide peer help. Since I-Help is designed to provide peer help, it does not provide any instructor support. As learner support in the CSCL environment, I-Help provides public and private forums for the users to discuss and post their thoughts. In addition to providing forums, I-MINDS also allows the learners to use whiteboards to discuss their ideas. As collaborative support, I-Help agents negotiate on behalf of the learners to find the most suitable peer who is willing to help. However, this group formation is one-on-one basis. On the other hand, I-MINDS provides group formation for groups of various sizes. Moreover, I-MINDS also monitors and analyzes performances of the groups.

Teixera et al. (2003) presented MATHNET, a multiagent CSCL environment where the students can learn by interacting and collaborating with the system and among themselves. MATHNET facilitates collaborative learning with tutor agents, pedagogical agents, learner modeling agents, and searching agents. The learner modeling agents model the learners, the searching agents select the appropriate learning material the learners, the pedagogical agents, and the tutoring agents provide the appropriate teaching strategy for the CSCL session. The instructor in MATHNET assumes an active role like one in I-MINDS and has the capability of monitoring and evaluating individual as well as group activities. For learner support, MATHNET provides tools that the learners can use to communicate with the system, their peers, their own group, and other groups by exchanging messages. For collaborative support, MATHNET forms groups that contain users with a common goal so that they can help each other to achieve that goal. In contrast, I-MINDS uses an adaptive and learning enabled group formation algorithm that dynamically forms groups based on the learners' profiles and their evaluation of the peers. Furthermore, I-MINDS also monitors and analyzes the performance of the groups.

Constantio-González et al. (2003) proposed a web-based environment called Collaborative Learning Environment for Entity-Relationship Modeling (COLER) in which students can solve Entity-Relationship (ER) problems while working synchronously in small groups at a distance. The learner support provided in the COLER system is similar to that provided in I-MINDS. COLER, like I-MINDS, includes a message exchange tool and a common digital whiteboard where the students work collaboratively. For instructor support, COLER allows the instructor to monitor and evaluate the

individual and collaborative work of the learners during and after collaboration. This is also provided as instructor support in I-MINDS. For collaborative support, COLER and I-MINDS use coaches and group agents, respectively, to monitor and evaluate the learners' collaborative performances as well as the performance of the groups. However, the groups in COLER are formed by the instructor whereas in I-MINDS, the groups are automatically formed based on the learners' profiles and their evaluation of other learners.

Barros and Verdejo (2000) proposed the Distance Environment for Group Experiences (DEGREE) architecture which is designed to analyze and characterize individual and group work of learners in a CSCL environment. Their aim is to determine the quality of the activities of the individual learners and the learner groups. Like I-MINDS, DEGREE provides similar functionalities for instructor support (e.g., assigning task, form groups, etc.). For learner support, DEGREE provides only text-based communication. Whereas, I-MINDS allows the learners to communicate using text messages and whiteboards. For collaborative support, DEGREE categorizes the messages exchanged among students into progressing stages of collaborative activities and analyzes each stage accordingly. This qualitative analysis is definitely more detailed than the text-based collaboration analysis provided by I-MINDS. However, unlike DEGREE, I-MINDS is equipped with an automated coalition formation algorithm.

Tedesco (2003) proposed a system called MARCo to utilize conflicts as triggers of cognitive changes in group interactions. To detect and mediate meta-cognitive conflicts, MARCo analyzes the ongoing interaction by taking into account: (1) the model of strategic changes, (2) the model of the group, and (3) the history of the interaction. Focusing on detecting and mediating cognitive conflicts, MARCo aims to invoke articulation and reflection in the learners to improve the quality of their solutions. Thus, MARCo lacks the functionalities supporting collaborative activities as does I-MINDS.

Vizcaíno (2005) describes a virtual, simulated student architecture that detects and avoids three situations that decrease the benefits of learning in collaboration: off-topic (off-task) conversations, students with passive behaviors, and problems related to students' learning. The learner support in their CSCL system is better than I-MINDS since, they provide guidance to the learners when they are not learning or talking off topic. However, they do not account for any instructor support in their system. Although the learner guidance provided by the simulated student translates to better group behavior, unlike I-MINDS, their system does not include any tools to create, monitor and evaluate behaviors of the groups.

We have summarized the above discussions in Table 7.

Table 7
Comparison of CSCL research projects and systems with I-MINDS

Research Projects	Learner Support	Instructor Support	Collaborative Support
KA-Agent (Ogata and Yano 2000)	Text-based communication functionalities, knowledge awareness	No	No
I-Help (Vassileva et al. 2002, Bull et al. 2001, Vassileva et al. 1999)	Public and private forums	No	Suitable peer matching
MATHNET (Teixera et al. 2003)	Text-based communication tools to interact with the system	Individual and group activities monitoring	Group formation of users with common learning goals

	and the peers		
COLER (Constantio-González et al. 2003)	Text-based communication, whiteboard	Individual and group activities monitoring	Evaluation of group performances of users
DEGREE (Barros and Verdejo 2000)	Text-based communication	Text-based interaction with the students	Qualitative analysis of the collaborative process
MarCo (Tedesco 2003)	Text-based communication	No	Detection of meta-cognitive conflicts by analyzing collaborative interaction
Simulated Student (Vizcaíno 2005)	Text-based communication and guidance to the learners when they are distracted (e.g., off-topic discussions)	No	No
I-MINDS	Text-based communication tools, whiteboard, and a tool to receive and archive slides from the instructor, agent based group formation	Text-based interaction with the students, ranking and scoring student questions, quizzes, evaluations, automated group formations, and monitoring of individual and group performances	Evaluation of individual and group performances after a collaboration session

CONCLUSIONS

A computer-supported cooperative learning system called I-MINDS has been proposed and described. The I-MINDS framework is based on a multiagent system where intelligent agents work behind-the-scenes to serve the instructors and students. These agents interact with each other to share information and learn over time to improve their performance. Presently, I-MINDS has a suite of multimedia capabilities and intelligent modules. Its agents facilitate group activities by forming compatible groups, support structured cooperative learning such as Jigsaw, help manage a classroom through evaluating and grouping questions, and learn to improve their heuristics and knowledge bases. I-MINDS has been deployed and evaluated in an introductory computer science course (CS1). Results show that I-MINDS can support cooperative learning effectively in place of face-to-face collaboration among students in weekly laboratory sessions. Further, the agent-based design of I-MINDS supports modular extension to the system. Finally, data collected by I-MINDS provided vital information on student group activities, showing that I-MINDS can also be an effective testbed for educational research.

Future work includes improvements of our group formation method, improvements of our modeling of group activities and improvements of the collaborative support provided to the learner groups. We are improving our group formation algorithm by improving our compatibility measuring technique of the potential group members. Currently, we use student perceptions of each other

measured by surveys to estimate the compatibility between them. Compatibility measures that we are considering include the learning goal ontology described in (Inaba et al. 2000) where students post their learning goals (“intend to learn by being taught”, “intend to learn from observation”, “intend to learn by teaching”, etc.) and students with compatible learning goals (e.g., “intend to learn by being taught” and “intend to learn by teaching”) are teamed up. We are also investigating dialogue analysis. Currently, I-MINDS does parse student messages only to categorize the messages for similarity, not for dialogue analysis. Dialog moves made by students such as pumps, positive pumps, splices, prompts, hints, summaries, elaborations, positive feedback, negative feedback, and neutral feedback, as used in AutoTutor (Person et al., 2001) could be used to label student contributions and roles in a conversation, for example. Pilkington (2001) pointed out that the goals of educational dialogue analysis are to identify the features that distinguish instructional discourse from other types of discourses and to determine what makes it effective. We are now working to integrate dialogue analysis with the student historical profiles performing as a team and as an individual currently available in the design. This will allow I-MINDS agents to better model a student or group in relation to what the student or group has performed in the past.

We are also investigating approaches to improve the user modeling accuracy of our agents. In particular, Introne et al. (2006) proposed the notion of Coordinating Representation (CR) that collects task and coordination-specific information about a group where intentions and common knowledge are shared among learners. Further, we are also reviewing the *Bootstrapping Novice Data* (BND) (Harrer et al. 2006) framework where actions can be recorded in a single graph called the Behavior Representation (BR) graph, allowing the system to follow the reasoning of the collaborating learners in order to support the activities better.

ACKNOWLEDGEMENTS

The authors would like to thank the National Center for Information Technology in Education (NCITE), NSF (SBIR grant DMI-044129) and Microsoft Research ConferenceXP (CX) for the research funding. The authors would also like to thank Xuli Liu, Phanivas Vemuri, Suresh Namala, and Xuesong Zhang for their programming work. The authors would also like to thank the anonymous reviewers for their comments and suggestions that significantly improved the paper.

REFERENCES

- Aronson, E., Blaney, N., Sikes, J., Stephan C., & Snapp, M. eds. (1978). *The Jigsaw Classroom*. Beverly Hills, CA: Sage.
- Barros, B., & Verdejo, M.F. (2000). Analysing student interaction processes in order to improve collaboration. The DEGREE approach. *International Journal of Artificial Intelligence in Education*, 11, 221-241.
- Beaudoin, M.F. (2002). Learning or lurking? Tracking the “invisible” online student. *The Internet and Higher Education*, 5(2), 146-155.
- Bull, S., Greer, J., McCella, G., Kettel, L. & Bows, J. (2001). User modeling in I-Help: what, why, when and how, in *Proc. 8th Int. Conf. on User Modeling*, Berlin, Germany, 117–126.
- Chalmers, C. & Nason, R. (2005). Group metacognition in a computer-supported collaborative learning environment, in *Proc. Int. Conf. on Computer Education (ICCE'05)*, Singapore, Singapore, 35-41.
- Chang, C.-C. (2003). Implications and issues of building a distributed Web-based learning community, in *Proc. Int. Conf. on Computers in Education (ICCE'2003)*, Hong Kong., China, 224-233.
- Clarke, J. (1994). Pieces of the puzzle: The Jigsaw method. In *Handbook of Cooperative Learning Methods*. Westport, CT: Greenwood Press.

- Constantio-González, M. A., Suthers, D. D., & Escamilla de los Santos, J. G. (2003). Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education*, 13, 263-299.
- Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R. and Tutoring Research Group. (1999). Autotutor: A simulation of a human tutor. *J. Cognitive Systems Research*, 1, 35-51.
- Harrer, A., McLaren, B. M., Walker, E., Bollen, L. and Sewall, J. Creating cognitive tutors for collaborative learning: steps toward realization. *User Modeling and User-adapted Interaction (special issue on user modeling to support groups, communities and collaboration)*, 16(3-4), 175-209.
- Hoppe, U. (1995). Use of multiple student modeling to parametrize group learning, in *Proc. Int. Conf. on Artificial Intelligence in Education (AIED'95)*, Washington, DC, 234-249.
- Huhns, M., N. and Stephens, L., M. (2000). Multiagent systems and societies of agents, in Weiss, G. (Eds.) *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*, Chapter 2, 80-81, MIT Press.
- Ikeda, M., Go, S., & Mizoguchi, R. (1997). Opportunistic group formation, in *Proc. Int. Conf. on Artificial Intelligence in Education (AIED'97)*, Amsterdam, Netherlands, 166-174.
- Inaba, A., Supnithi, T., Ikeda, M., Mizoguchi, R., and Toyoda, J. (2000a). How can we form effective collaborative learning groups?, in *Proc. Int. Conf. on Intelligent Tutoring Systems (ITS)*, Montreal, Canada, 282-291.
- Introne, J., and Alterman, R. (2006). Using shared representations to improve coordination and intent inference. *User Modeling and User-adapted Interaction (special issue on user modeling to support groups, communities and collaboration)*, 16(3-4), 249-280.
- Johnson, D. & Johnson, R. (1999). Making co-operative learning work. *Theory into practice*, 38(2), 67-73.
- Khandaker, N. (2005). *VALCAM: An Auction Based Learning Enabled Multiagent Coalition Formation Algorithm for Real-World Applications*, M.S. Thesis, Computer Science & Engineering, University of Nebraska, Lincoln, NE.
- Liu, X., Zhang, X., Soh, L.-K., Al-Jaroodi, J., & Jiang, H. (2003b). A distributed, multiagent infrastructure for real-time, virtual classrooms, in *Proc. Int. Conf. on Computers in Education (ICCE2003)*, Hong Kong, China, 640-647.
- Liu, X., Zhang, X., Al-Jaroodi, J., Vemuri, P., Jiang, H., & Soh, L.-K. (2003a). I-MINDS: an application of multiagent system intelligence to on-line education, in *Proc. IEEE Int. Conf. Systems, Man, & Cybernetics*, Washington, D.C., 4864-4871.
- Mitchel, T., M. (1997). Introduction, in *Machine Learning*, Chapter 1, 1-5, McGraw Hill.
- Mulder, I., Swaak, J. & Kessels, J. (2002). Assessing group learning and shared understanding in technology-mediated interaction. *Education Technology & Society*, 5(1), 35-47.
- Namala, S. (2004). *An intelligent module for I-MINDS*. M.S. Project Report, Computer Science & Engineering, University of Nebraska, Lincoln, NE.
- Ogata, H., Matsuura, K., & Yano, Y. (1996). Sharlok: Bridging learners through active knowledge awareness, in *Proc. IEEE Int. Conf. on Systems Man & Cybernetics*, Vol 1, 601-606.
- Ogata, H., & Yano, Y. (2000). Combining knowledge awareness and information filtering in an open-ended collaborative learning environment. *International Journal of Artificial Intelligence in Education*, 11, 33-46.
- Olney, A., Louwerse, M., Mathews, E., Marineau, J., Hite-Mitchell, H., & Graesser, A. (1995). Utterance classification in Autotutor, in *Proc. HLT-NAACL Workshop*, 1-8. Philadelphia: Association for Computational Linguistics.
- Person, N. K., Graesser, A. C., Kreuz, R. J., Pomeroy, G. And the Tutoring Research Group (2001). Simulating human tutor dialogue moves in AutoTutor. *International Journal of Artificial Intelligence in Education*, 12, 23-39.
- Picciano, A. G. (2002). Beyond Student Perceptions: Issues of Interaction, Presence, and Performance in an Online Course. *Journal of Asynchronous Learning Networks*, 6(1), 21-40.
- Pilkington, R. (2001). Analyzing educational dialogue interaction: towards models that support learning, *International Journal of Artificial Intelligence in Education*, 12, 1-7.

- Sandholm, T. (2000). Auctions, in Weiss, G. (Eds.) *Multiagent Systems: A Modern approach to Distributed Artificial Intelligence*, Chapter 5, MIT Press, 211-219.
- Sekine, S., & Grisman, R. (1995). A corpus-based probabilistic grammar with only two non-terminals, in *Proc. 4th International Workshop on Parsing Technologies*, Prague, Czechoslovakia, 216-223.
- Sen, S. and Weiss, G. (2000). Learning in multiagent systems, in Weiss, G. (Eds.) *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*, Chapter 6, 261, MIT Press.
- Soh, L.-K. (2004). On cooperative learning teams for multiagent team formation, *Technical Report WS-04-06 of the AAI'2004 Workshop on Forming and Maintaining Coalitions and Teams in Adaptive Multiagent Systems*, San Jose, CA, 37-44.
- Soh, L.-K. (2006). Implementing the Jigsaw model in CS1 closed labs, in *Proc. 11th Annual SIGCSE Conf. on Innovation and Technology in Computer Science Education (ITiCSE'2006)*, Bologna, Italy, 162-167.
- Soh, L.-K., Jiang, H., & Ansoorge, C. (2004b). Agent-based cooperative learning: a proof-of-concept experiment, in *Proc. 35th Technical Symposium on Computer Science Education (SIGCSE'2004)*, March 3-7, Norfolk, VA, 368-372.
- Soh, L.-K., Khandaker, N., Liu, X., and Jiang, H. (2005b). Computer-supported structured cooperative learning, in *Proc. Int. Conf. on Computers in Education (ICCE'2005)*, Singapore, 428-435.
- Soh, L.-K., Khandaker, N., Liu, X., & Jiang, H. (2006a). A computer-supported cooperative learning system with multiagent intelligence, in *Proc. Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'2006)*, May 8-12, Hakodate, Japan, pp. 1556-1536.
- Soh, L.-K., Khandaker, N., & Jiang, H. (2006b). Multiagent coalition formation for computer-supported cooperative learning, in *Proc. Innovative Applications of Artificial Intelligence (IAAI'2006)*, Boston, MA, 1844-1851.
- Soh, L.-K., Liu, X., Zhang, X., Al-Jaroodi, J., Jiang, H., & Vermuri, P. (2004a). I-MINDS: an agent-oriented information system for applications in education. *Lecture Notes in Artificial Intelligence (LNAI 3030) Series Agent-Oriented Information Systems*, Springer-Verlag, 16-31.
- Soh, L.-K., Zhang, X., Liu, X., & Jiang, H. (2005). Intelligent collaborating agents to support teaching and learning, in PytlíkZillig, M. Bodvarsson, and R. Bruning (Eds.) *Technology-Based Education: Bringing Researchers and Practitioners Together*, Chapter 10, 203-224, Information Age Publishing.
- Tedesco, P. (2003). MARCo: Building an artificial conflict mediator to support group planning interactions. *International Journal of Artificial Intelligence in Education*, 13, 117-155.
- Teixeira, C. Labid, S. and Nascimento, E. (2002). Modeling the cooperative learner based on it's actions and interactions within a teaching-learning session. *Frontiers in Education*, 1, 19-23.
- Vassileva, J., McCalla, G., & Greer, J. (2002). Multi-agent multi-user modeling in I-Help. *User Modeling and User-Adapted Interaction: Special Issue on User Modeling and Intelligent Agents*, 13(1), 1-31.
- Vassileva, J., Greer, J., McCalla, G., Deters, R., Zapata, D., Mudgal, C., & Grant, S. A. (1999). A Multiagent design of a peer-help environment, in Lajoie, S. & Vivet, M. (eds), *Artificial Intelligence in Education*, 38-45, IOS Press.
- Vemuri, P. (2003). *Multimedia support for I-MINDS*. M.S. Project Report, Computer Science & Engineering, University of Nebraska, Lincoln, NE.
- Vizcaíno, A. (2005). A simulated student can improve collaborative learning. *International Journal of Artificial Intelligence in Education*, 15, 3-40.
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2), 115-152.
- Zhang, X. (2004). *I-MINDS: An intelligent multiagent system supported teaching and cooperative learning environment*. M.S. Thesis, Computer Science & Engineering, University of Nebraska, Lincoln, NE.
- Zhang, X., Soh, L.-K., Jiang, H., & Liu, X. (2005). Using multiagent intelligence to support synchronous and asynchronous learning. *Studies in Fuzziness and Soft Computing*, Volume 178, May 2005, 111-140.