A Distributed Software System Architecture For Wireless Peer-to-peer Collaborative Learning

Indrani Vedula and Richard Han
University of Colorado
Department of Interdisciplinary Telecommunications, Department of Computer Science
vedula@colorado.edu, rhan@cs.colorado.edu

Abstract

Students often turn to their peers for help in order to learn a new concept or lesson introduced by a teacher in class. This establishes roles of tutor and learner between students, which can also reverse depending on the subject, with the teacher as a third-party mediator. This paper discusses our design of a distributed software system architecture that seeks to harness the tutor-learner relationship between peers into a collaborative learning system. Our goal is to provide a verifiable, portable, and inexpensive system of coordinated wireless handhelds that both promotes learning of lesson plans by the students and enhances the tutoring skills of students.

1. Introduction

Students often turn to their peers for help in order to learn a new concept or lesson introduced by a teacher in class. The smartest student peers are often consulted as tutors to their other peers. These roles of tutor and learner can also reverse depending on the subject matter being studied, i.e. a student that is adept at languages may not be as adept at mathematics, and vice versa. In more formal settings, certain students are assigned to formally be tutors, and in this case the teacher serves as a third-party mediator that measures and verifies the progress of each tutor-learner pair. This type of collaborative learning via peer-based tutorial behavior occurs at all levels of the educational system, from grade school up through college and beyond.

Whether students consult their peers informally or formally to understand a new lesson, the benefits of the tutor-learner system are three-fold: first, the student learner gains an improved understanding of a lesson that wasn't clear from the teacher alone; the tutor also becomes a more effective (student) teacher through the process of tutoring; and the teacher benefits because the students have additional resources, namely their ablest peers, for consulting and grasping a new concept or lesson.

To harness these mutual educational benefits, we have designed a distributed software architecture whose aim is to capture the essential collaborative learning aspects of this peer-based tutorial behavior. Such an architecture should be able to verify that a student has actually learned a new concept or lesson, or measure to what extent a student learner has grasped a concept. Our architecture should also be able to verify that a student tutor has played a quantifiable role in helping the learner to achieve an improved understanding, i.e. our system should be able to measure that the tutor has improved his/her abilities as a student teacher. These two properties are the critical foundations of peer-based collaborative learning: both sides improve their skills through the tutor-learner relationship.

Other models of collaborative learning commonly found on the Internet and in personal computing tend to be lacking in benefiting both sides of a collaboration or require modification for our educational environment. For example, students can consult class newsgroups or bulletin boards on the Web to seek expertise in answering a question that they haven't understood from class. These loosely structured cyber communities represent client-server paradigms for learning. These communities do not typically measure the quality of the expertise given, nor evaluate whether the so-called expert improves their advice over time. Certain Web sites such as www.experts-exchange.com do evaluate the expertise of the experts over time. Users can pose questions and assign a point value to the difficulty of questions, and when accepting answers from experts can assign a grade to the quality of the answers. Experts accumulate points for each correct answer based on the difficulty assigned to the question and the grade assigned to the answer,

Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)
0-7695-1967-9/03 $17.00 © 2003 **IEEE**

leading to hall-of-fame experts. We will leverage this approach for our design, but will modify it to suit our educational context. For example, rather than have the learners grade the answers of the tutors, as would be suggested by the above approach, we instead introduce an omniscient third-party mediator, namely the teacher, who grades the quality of the answer emerging from the tutor-student collaboration. We also considered the paradigm wherein the personal computer acts as the tutor, but quickly dismissed this approach as non-collaborative, i.e. there is no means for developing the human tutoring skills of the students, and hence no mutually beneficial collaborative learning that takes place.

A design goal of our architecture was to accomplish collaborative learning inexpensively and in a portable manner. As a result, our focus has been to develop our system over a distributed collection of wireless handhelds. Prior studies have been aimed at developing a collaborative learning environment across various platforms. Bilezikjian et al [1] created a collaborative application called Geney on handheld computers where participants shared a screen information view. Roschelle et al [2] incorporated an interactive model between the teacher and the student and also provided an assessment of their performance in the form of a statistical histogram. These applications were developed on a framework termed as Wireless Internet Learning Devices (WILD). Kam et al [3] developed a classroom learning technology called Livenotes that facilitated peer learning through a shared whiteboard that aided in collaborative note taking, and included wireless handheld computers. A similar application that has been developed by Kusunoki et al. [4] integrates a board game and a computer simulation for studying urban planning and environmental issues. The collaboration between the students increased by representing the learning system as a game. The Concord Consortium has developed an interesting collaborative teaching-learning model that allows the teacher to track the student's progress by posing a question on the student's handhelds thus testing and monitoring the individual student's understanding [5]. The Social Net application uses handheld wireless RF communication devices called Cybikos for studying patterns of common interest between users and enhances their interaction [6]. This highlights the capabilities of Cybikos [7] as effective learning devices, and we will use these devices in our implementation. Algebra Jam [8] is an effort to provide a multi-student team tutoring or collaborative system. Our focus is instead on pairwise tutoring. Mcdonald et al discuss an Expert Recommender system that provides a general architecture for locating expertise [9].

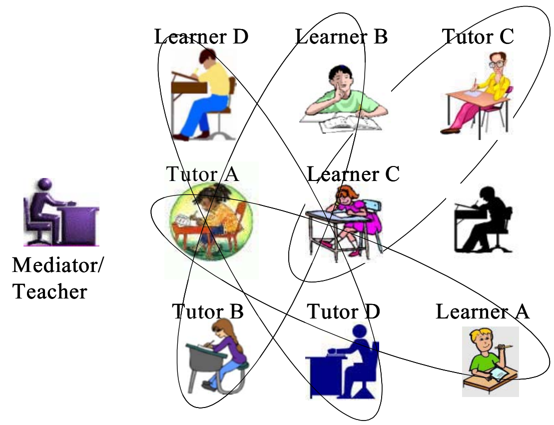

Figure 1. Tutor-learner pairs and a classroom mediator.

In section II we present the system architecture of our application. We describe our experimental test bed in section III, present our analysis about this system in section IV, and discuss future work in section V.

2. System Architecture

The architecture of our wireless peer-to-peer collaborative learning system consists of 3 basic entities, namely a Learner, a Tutor, and a Mediator, as shown in Figure 1. The learner is a student who is given a problem to solve. The tutor is another student that guides the learner towards the correct solution of the problem. The role of a mediator in our system is to pose the question to the tutor and the learner and to supervise the dialogue between the two entities. The wireless protocol that governs interaction between these entities is described in the following subsections.

2.1. Roles

The duties of the learner are to solve the problem posed to the learner by the mediator on the learner's handheld device. The learner can request help from the tutor in the form of hints to obtain the right approach towards the solution. The learner is also responsible for sending the proposed solution across to the mediator within a fixed time limit. The learner earns points for each correct answer. If enough points are accumulated, the learner can eventually be chosen to be a tutor for this particular subject. In this way, our system organically evolves to identify the ablest students per topic and migrates the ablest students towards roles as tutors.

The tutor's role is to guide the learner. In our system, the learner requests hints from the tutor. The tutor earns points for each correct "assist", i.e. for each correct answer that the learner sends back to the mediator. Therefore, the tutor has an incentive to provide the best hint possible to maximize the chances that the learner grasps the concept and comes up with the correct answer. In this way, the teaching skills of the tutoring student will be enhanced. Our aim is to capture typical tutorial behavior when a student says "I don't understand how to solve 2+2=__. Can you give me a hint?". In this case, a tutor often explains the approach in one way, and if the learner still doesn't understand, the tutor tries another way of explaining the same concept. The tutor is allowed to provide multiple hints.

The mediator's duties include forming and posing the question to both the learner and the tutor. This entity is also responsible to make sure that the learner and the tutor are learning and improving their skills and are not cheating while exchanging ideas. To prevent cheating, our system logs all electronic dialogue between the tutor and learner, allowing the mediator to monitor learning and therefore verify that true progress has occurred for both the learner and tutor. Since our system is designed to be operated within a classroom of about thirty students, then proctoring of the classroom, most likely by the mediator/teacher, will be sufficient to prevent cheating through spoken/visual communication.

2.2. Communication protocol

Figure 2 describes the communication protocol between the learner, tutor, and mediator. In step (1), the mediator selects a problem or question and downloads both the problem and recommended hints to the tutor. Our software also downloads the network address of the learner's handheld, so that the tutor and learner can communicate directly. However, the identity of the learner is obscured from the tutor, and vice versa. This protects anonymity in both directions. In step (2), the mediator downloads just the problem statement to the learner as well as the network address of the tutor, though not the identity of the tutor. At this stage, the learner and tutor can effectively communicate with each other directly. If the learner is unable to answer the question immediately, then the learner can request a hint from the tutor in step (3). The tutor responds in step (4) with a well-chosen hint. The tutor may devise his/her own hint, rather than use the recommended hints. By crafting their own clever hints, a tutor will be able to advance his/her teaching skills. If instead the tutor selects the best hint from the list of recommended hints, the tutor will still be exercising their best judgment and thereby enhance their teaching skills. After the tutor responds with a hint, the learner may choose to submit an answer in step (5). The learner and the tutor are given a fixed amount of time to solve the problem collaboratively. Once the time is up, the learner is required to send the answer to the mediator.

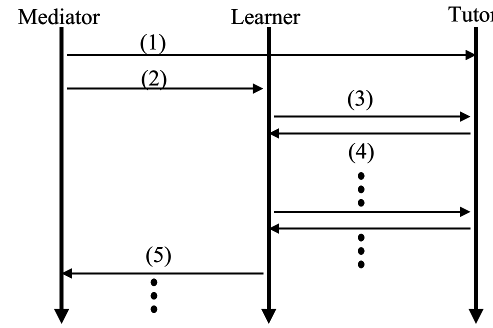

Figure 2. Communication protocol.

2.3. Evaluating student performance

The mediator scans the logs received to ensure that the communication between the pairs is cheat-proof. To evaluate the performance of each student, the mediator awards points to the learner for each correct answer and "assist" points to the tutor for each correctly tutored answer. A student may answer without requesting help from the tutor. In such a case, the tutor is awarded no "assist" points. A tutor may offer unrequested hints. Since the mediator logs all dialogue between learner and tutor, then the mediator can also detect if there has been excessive hint-giving from the tutor, or excessive requests for hints from the learner. In either case, our architectural design affords the mediator with an opportunity to penalize the appropriate party, either the learner or tutor, by deducting points. In this way, wrong answers can negatively affect the accumulated score of tutors and learners. A student who has momentarily been designated a tutor can, after a series of poor "assists" be demoted to learner, and conversely a learner can be promoted to tutor. Thus, there is constant incentive to improve on both the parts of the learner and tutor. Over time, the students with the highest aggregate number of correct answers are migrated towards the role of tutors. This is an egalitarian philosophy. Rather than grouping all smart students together, the ablest students are leveraged to assist the weaker students.

After a successful answer, our system offers the flexibility of switching tutors and learners immediately, or continuing with the current pair. If the students are to switch, then the two students go back into the pool until such time as another pair is freed up for anonymous

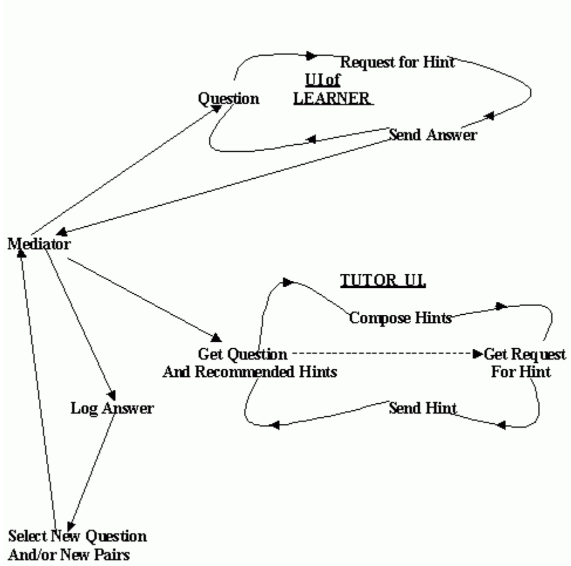

Figure 3. State diagram of UI.

rematching. Preserving anonymity between the learner and tutor is designed to conceal any personal differences among the students, or any social stigmas that may arise from requesting help or even giving help. As shown in Figure 1, students in a classroom will be unable to tell who has been assigned to be either their tutor or learner. Also, the scores of students can be kept private at the mediator's discretion.

3. Experimental test-bed

To create a working prototype of the proposed system, we selected Cybiko wireless handhelds as the foundation of our system. A database server logged the exchanges between the entities and their current scores. The Cybiko handhelds form a peer-to-peer wireless network with a range of 150- 300 feet. These devices are highly cost effective, at about $70 a piece, and offer a simple development platform. They have been classified as effective learning devices.

The basic protocol code was written in C and a cross compiler for Cybikos called “vcc” was used to compile these programs. The application created was loaded on to the Cybiko devices through a USB port using software called Cyloader.

A state diagram of the UI’s at the learner, tutor, and mediator and their interactions is shown in the Figure 3. Each UI consists of multiple screens of menus. For example, for the learner, there is a Main menu where a question can be obtained, a hint can be requested, and a

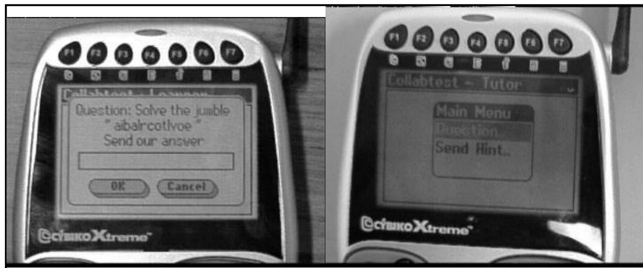

Figure 4. (a) Learner's UI. (b) Tutor's UI.

solution can be sent to the mediator. Snapshots of representative screens for the learner and tutor are shown in Figure 4. For example, in Figure 4(a), the learner has downloaded a jumble question that the tutor must assist in solving with hints. We also downloaded arithmetic and grammar questions.

4. Analysis

We conducted experimental test sessions with a single tutor-learner pair who had no knowledge of the system. It was found that the initial reaction of the users was to browse through all the menu options of the application on the screen even before the mediator broadcast the question. This indicates that, ideally the menu options of the application should be locked until the mediator transmits the question. We also observed that it was rather difficult to convey a hint electronically. This was because of the User Interface limitations of the Cybiko device, which can be considered to lack richness as compared to other user interfaces that can be created on other handhelds or laptops. The task of formulating a hint is a tougher process since they need to be very explicit and short. It was also noticed that learners who were smart would answer the questions without requesting any hint from the tutors. For effective collaborative learning, the questions should be designed such that an average student as a learner would require interaction with the tutor to answer the question. A final observation is that such a collaborative learning system is most effective when the questions formulated by the mediator are objective in nature requiring an analytical thought process to get to the correct solution. Examples of such questions are in the area of mathematics, physics, english grammar and analytical/logical reasoning.

5. Discussion and Future Work

Our initial experiments lack the rigorous testing that a real-world classroom environment would provide to evaluate the performance of the proposed collaborative learning system. As a result, we are keenly interested in pursuing a classroom implementation of this electronic peer-to-peer learning concept. Such a deployment would enable us to answer several intriguing questions. For example, to what extent should anonymity be preserved in a tutor-learner relationship? Just as consumers consult trusted and well-known experts such as Consumer Reports and AAA ratings to evaluate products and services, students often prefer to consult a well-known and trusted student whose expertise in a given subject has already been clearly demonstrated. By preserving anonymity, our system hides the identity of the trusted expert and requires the learner to trust the expertise of the unknown tutor. Particularly when the system is first initialized, tutor-learner pairs will be assigned randomly, resulting in pairings in which the learner may be more capable than the anonymous tutor. If such role reversals occur too often, then the learners may very well stop trusting both the system and the hints provided by the "so-called" tutor.

A related issue is how to optimally evolve the system so that the effectiveness of collaborative learning is maximized. After a pair of students has answered a question correctly or incorrectly, should the pair be given another chance, another question, or should the roles of tutor and learner be reversed? After how many questions should the two students be reassigned to new tutor-learner pairs? Our hope is to explore these issues within a classroom evaluation framework.

6. Conclusions

We have developed a peer-to-peer collaborative learning system with an evaluation strategy for the learner's progress. Our software architecture is distributed between a mediator and multiple tutor-learner pairs. Our system provides a highly flexible verifiable means of measuring the learning progress of both student learners and student tutors. This system architecture has been implemented on wireless Cybiko Xtreme handhelds.

7. Acknowledgements

We wish to thank SheetalKumar Doshi for his help with the experiments.

8. References

[1] M.Bilezikjian, K.Inkpen, Regan L, S. Klemmer and J. Landay, "Exploring a new interaction paradigm for collaborating on handheld computers", UC Berkeley Computer Science Division Technical Report, November 2000. http://guir.berkeley.edu/projects/geney/GeneyTechReport.pdf

Figure 5. Testing sessions with students.

[2] J. Rochelle and R. Pea, "A walk on the WILD side: How wireless handhelds may change CSCL", Proceedings of CSCL'02. http://newmedia.colorado.edu/cscl/79.pdf

[3] M. Kam, J. Canny, A. Illes and D. Glaser, "Learning via distributed dialogue: LIVENOTES and Handheld Wireless Technology", Proceedings of Conference on Computer Support for Collaborative Learning, Boulder, Colorado, January 2002. http://newmedia.colorado.edu/cscl/225.html

[4] F.Kusunoki, M.Sugimoto, H. Hashizume, " A System for Supporting Group Learning that Enhances Interaction", Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 Conference (pp. 323-327). http://kn.cilt.org/cscl99/A40/A40.HTM

[5] C.Staudt, "Handhelds track student progress" http://www.concord.org/newsletter/2002winter/handhelds.html, Accessed on Feb 08, 2003.

[6] M.Terry, E.D.Mynatt, K.Ryall, D.Leigh, "Social net: using patterns of physical proximity over time to infer shared interests", CHI'02 extended abstracts on Human Factors and Computing Systems. http://www.cc.gatech.edu/grads/t/Michael.Terry/papers/social_net_CHI.pdf

[7] Cybiko, ***http://www.cybikoxtreme.com/***, Accessed on Feb 08, 2003.

[8] M.K.Singley; P.G.Fairweather; S.Swerling," Team Tutoring Systems: Reifying Roles in Problem Solving", Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 Conference (pp. 538-548). http://www.research.ibm.com/AppliedLearningSciWeb/Singley/Singley1.htm

[9] D. McDonald and M. Ackerman, "Expertise recommender: a flexible recommendation system and architecture", Computer Supported Cooperative Work (CSCW), 2000, pp. 231-240.