Implementing the Jigsaw Model in CS1 Closed Labs

Leen-Kiat Soh
National Center for Information Technology in Education (NCITE)
Department of Computer Science and Engineering
University of Nebraska
256 Avery Hall, Lincoln, NE 68588-0115 USA
E-mail: lksoh@cse.unl.edu

ABSTRACT

We apply the Jigsaw cooperative learning model to our CS1 closed labs. The Jigsaw cooperative learning model assigns students into main groups in which each group member is responsible for a unique subtask, gathers all students responsible for the same subtask into a same focus group for focused exploration, returns all students to their original main groups for reporting and reshaping, and then each group integrates the solutions for the subtasks from its members. For our study, we used the Jigsaw model in three CS1 closed labs. For each, there were three sections: (1) students worked individually, (2) students worked in groups using Jigsaw, and (3) students worked in groups using a computer-supported Jigsaw environment. The post-test scores of the three sections are compared to study the impact of Jigsaw and the feasibility of using a computer-supported Jigsaw design. Further, we investigate how the three lab topics (debugging, unified modeling language (UML), and recursion) affected impact of Jigsaw model on student performance

Categories and Subject Descriptors
K.3 [**Computers and Education**]: Computer Uses in Education, Computer Science Education

General Terms
Design, Experimentation

Keywords
Closed Laboratories, CS1, Cooperative Learning, Jigsaw

1 INTRODUCTION

Cooperative learning is "instruction that involves students working in teams to accomplish a common goal." [7]. For computer science education, various forms of cooperative learning have been used and found to lead to improve student learning, motivation, and performance. Techniques such as pair programming (e.g., [4, 6, 11, 19]) and game-based learning (e.g., [9, 12, 15]), have also been reported to have improved CS instruction and learning. In pair programming, two students collaborate to produce a program that solves a problem. In game-based learning, teams of students could participate in a game with winning the game being the objective of each team, or individual students could participate in a game with winning the game being the objective of each student. In the latter scenario, individual students, though not working in teams, could interact with their fellow classmates or observe and learn from the performance of their classmates during the game.

One of the underlying components that benefit students through cooperative learning is *learning by teaching* [6]. In learning by teaching, a student A teaches a student B about a particular subject. Not only B learns from this process, but A also learns from the articulation effort as teaching process forces A to explain explicitly what he or she understands cognitively to B [2]. Though learning by teaching has been used successfully in math, science, and humanities subjects (e.g., [13, 14]), explicit implementation of the learning by teaching approach has not been widely reported in CS instruction.

This paper reports on applying the Jigsaw cooperative learning model to CS1. The Jigsaw cooperative learning model was first introduced by Aronson *et al.* [1]. This procedure works as follows. First, the instructor assigns the students into groups. Second, the instructor divides a problem into different parts (or subtasks). Third, the instructor assigns a subtask to every student such that members of the same group will have different subtasks to solve. The students who are responsible for the same subtask then work together to come up with solutions to the subtask to which they have been assigned and develop a strategy for teaching the solutions to their respective group members. Clarke [3] further refined the Jigsaw structure into 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 subtask that they have been assigned, (3) ***Reporting and Reshaping***: The students return to their original groups and instruct their teammates based on their findings from the focus groups, and (4) ***Integration and Evaluation***: The team connects the various pieces generated by the individual members, address new problems posed by the instructor, or evaluates the group product. As we can see from the above definitions, Jigsaw's third phase "Reporting and Reshaping" involves explicit teaching by learning activities—the student who returns to his or her original group has to teach the group what he or she has learned from focused exploration.

Note that Johnson *et al.* [8] ranked the Jigsaw procedure ranked fourth, out of ten models, in terms of the cooperative versus competitive comparison. However, the impact of cooperative lessons was compared with competitive (e.g., game-based) learning, the Jigsaw model is the better one out of the only two models—evaluated in the study—that use specific *structures* in their respective designs. It was also noted that all the cooperative learning methods, including Jigsaw, are effective in increasing achievement [8]. Thus, we suspected that Jigsaw would be an appropriate model for our CS1 cooperative, closed labs that have structured activities.

Our study reported in this paper has three objectives. First, we want to investigate how the Jigsaw procedure impacts student learning and performance in CS1 closed labs. Second, we want to investigate how online Jigsaw groups fare as compared to in-person Jigsaw groups. Third, we want to investigate the suitability of different CS1 closed lab topics for the Jigsaw procedure. To support the online Jigsaw group, we utilize a computer-supported cooperative learning system called I-MINDS [10]. I-MINDS is a multiagent system that facilitates real-time, online collaboration among students.

In the following, we first describe the I-MINDS system and our CS1 closed labs. In Section 3, we describe the three laboratory topics chosen for our study and the modifications we made to I-MINDS to support the Jigsaw procedure. Then, we present our research method in Section 4. We discuss the results and our investigations in Section 5. Finally, we conclude.

2 BACKGROUND

2.1 The I-MINDS System

The I-MINDS system is a computer-supported cooperative learning system built on an intelligent multiagent technology [10]. Briefly, in I-MINDS, each student has a student agent while the instructor has a teacher agent. These agents interact with their respective users as well as among themselves. These agents exchange information, coordinate their actions, and track inter-agent activities behind the scene. Currently, a teacher agent, for example, automatically ranks student questions and sorts them for the instructor. A student agent, for example, automatically seeks out compatible student agents to invite to be the buddies of its user.

I-MINDS had previously been pilot-tested in a controlled experiment to assess what impact it had on student learning of Global Information Systems (GIS) content [16]. Results for the two testing sessions were encouraging. It was observed that the amount that the I-MINDS group improved from the pretest to the posttest was nearly twice that of the control group. The instructor using I-MINDS also noted that questions asked of him via I-MINDS tended to reflect a deeper understanding and demand a richer response than those questions posed during the control sessions.

I-MINDS has full multimedia capability—real-time audio and video streaming such that each student/instructor is able to transmit audio and video through the I-MINDS server. The I-MINDS agents are also equipped with (1) tracking capabilities—recording the messages communicated between student agents (or students), the length of each message, the time stamp of each message, each question asked by a student to an instructor or another student, and so on, and (2) collaborative features such as a chatroom and a digital whiteboard. In our study, we did not use the multimedia capability since the instructor and the students were in the same physical lab. We did use the tracking and collaborative features in our online Jigsaw treatment group.

2.2 Our CS1 Closed Labs

For our CS1 course, there are 14 weekly 2-hour closed labs. During each lab, students are required to carry out programming and problem solving activities specified in a handout. For each activity, students are required to answer several questions on a worksheet. Before the students are allowed to attend the lab, each is required to pass an online pre-test (>= 80%) which each is allowed to take as many times as necessary. At the end of the lab, each student is required to take an online post-test once. The students are scored individually for their worksheet answers and post-tests. The sum of all lab scores for the entire semester counts about 25% towards the final course grade for each student.

For each lab, the lab instructor briefly introduces the lab activities, and when necessary (e.g., such as teaching students about topics that are not covered in the course lectures) teaches through examples before letting the students start with the lab activities. Once the activities start, the lab instructor serves as a monitor and helper, going from one student (or group) to another to handle requests for help.

Our CS1 closed labs have been designed to resemble labs in physics, chemistry, and biology, where there are activities that require students to explore and experiment, in order to answer the questions posed on the worksheets.

Please refer to [17] for a discussion on our labs.

3 STUDY SETUP

Here we describe our choice of three laboratories out of 14 for our study and the modifications that we made to I-MINDS to support online Jigsaw activities in our study.

3.1 Laboratory Topics

We selected (1) Debugging and Testing, (2) Unified Modeling Language (UML), and (3) Recursion because these three labs are rather different. First, recursion is a conceptually difficult topic for students to learn while debugging and testing is more of a hands-on problem solving topic. The UML lab covers activities that reinforce the object-oriented programming notion in students while exposing students to a common software engineering practice. Second, while recursion is covered in the lectures, debugging and UML are not. Debugging and testing are not explicitly taught in lectures; students are taught debugging and testing in the labs, are familiar with them when completing their programming homework assignments, and exposed to testing in lectures (whenever the course instructor demonstrates programming concepts through “what-if” exercises).

Here below we briefly describe these labs. For details on these labs, please visit http://www.cse.unl.edu/reinventCS.

Debugging and Testing The objective of this lab topic is to provide hands-on experience to the students in (1) using a variety of debugging strategies to identify bugs in programs, (2) distinguishing between a syntax error and a semantic error, and (3) using a debug flag to enable/disable debugging information. There are four activities, constituting to four subtasks for this overall task of debugging and testing. The first activity involves debugging two programs by commenting out parts of the code. This is to fix syntax errors that cause the program to not compile. The second activity involves debugging two programs using a debug flag and additional `println()` statements. This is to demonstrate how to focus in on a bug and fix the bug such that the programs generate the correct output. The last two activities involve several different programs using the debugging strategy of a student's choice.

Unified Modeling Language (UML) The objective of this lab topic is to expose students to Unified Modeling Language (UML) such that they are able to (1) analyze the requirements that describe a simple application and use this information to design a set of classes that accurately reflect the requirements, (2) create a UML class diagram, and (3) create and document use cases to help analyze and design a simple application. There are two activities. The first activity is an instructor-led exercise, introducing students to the concept of UML by working through an example (designing a student registration system). The second activity is to design a resource management system. Within this activity, which is the overall task for the students, there are two sub-activities. Each group is given the system requirements for the to-be-designed resource management system. The first sub-activity is to analyze the written requirements, identify the nouns and verbs in the requirements, convert them into classes and associations/relationships between classes in the UML class diagram, and update the class diagram with class data values and methods. The second sub-activity is to analyze the written requirements, identify the actors who are either humans (groups) or any other systems that interact with the resource management system, identify the use cases by listing the functionality or services provided, diagram the use cases and draw the associations/relationships between use cases, and document the use cases. Thus, we use these two sub-activities as two subtasks for the Jigsaw procedure.

Recursion The objective of this lab topic is to provide hands-on programming experience to students such that they are able to (1) identify a recursive method, (2) identify the basic elements (i.e., stop conditions, end cases, integration step, recursive step) of a recursive method, (3) determine when a problem should be solved using a recursion, and (4) given a recursive mathematical definition for a problem, write a recursive Java method to solve the problem. There are three activities contributing to three subtasks of this overall task of learning about recursion. The first activity requires the students to check for a prime number based on a recursive algorithm. The second activity requires the student to find a file on a hard drive using a recursive solution. The final activity involves converting a recursive program to an iterative program.

3.2 Modification of I-MINDS

Though I-MINDS had been designed to support cooperative learning, it did not support *structured* cooperative learning like the Jigsaw procedure. Thus, we modified I-MINDS to do the following. The instructor is able to enter a task and list a set of subtasks, and announce this task to all students logged in to the I-MINDS environment. Based on the students' previous performance on collaboration tracked by I-MINDS—for the first lab in our study, all students were given an initial performance rating, the I-MINDS teacher agent automatically assigns the students into main groups such that each main group has a good mixture of high-performing and low-performing students. Then the instructor can announce the start of the Focused Exploration phase, upon which the teacher agent automatically assigns students to different subtasks for each main group, essentially identifying the members for each focus group. Note that the number of focus groups is the same as the number of subtasks. Once this phase starts, I-MINDS blocks students from sending messages to members other than their focused group members. The instructor can set a time limit on each phase as well. When the phase concludes, I-MINDS automatically brings every member back to their original main group (virtually), allowing them to send messages now within their respective main groups. Please refer to [18] for a detailed discussion of our modification of I-MINDS.

4 RESEARCH METHOD

In Spring 2005, we conducted our study in CS1. As indicated earlier, out of 14 weekly labs, we chose three labs for the experiments: debugging/testing, UML, and recursion. We had three lab sections with 17, 12, and 12 students, respectively. Lab section 1 served as the control lab—where the students worked individually on the three lab topics. Lab section 2 is the in-person Jigsaw treatment lab where students cooperated according to the Jigsaw model. In this lab section, the Introduction phase was first conducted. Then, the lab instructor announced the subtasks, and then assigned the students into several main groups. Then, the subtasks were assigned and the focus groups formed. Lab section 3 is the online Jigsaw treatment lab where students cooperated following the Jigsaw model using the modified I-MINDS. In this section, the students stayed at their individual computer stations and could only communicate through the chatroom and digital whiteboard of I-MINDS. The students were closely monitored and not allowed to communicate to each other verbally face-to-face.

For the other eleven lab topics, students in lab section 1 also worked individually while students in the other two sections worked in groups of three without any structure or the Jigsaw procedure. Thus in these cooperative activities, though the students had a chance to learn by teaching, they were not explicitly required to do so as opposed to what occurred during the Reporting and Reshaping phase for the Jigsaw treatment groups during the chosen three lab topics. Table 1 summarizes the setup.

Table 1. Lab sections and procedures.

Lab Section	**11 Labs**	**3 Selected Labs**
1 – Control	Individual	Individual
2 – In-Person Jigsaw Treatment	3-person groups	In-Person Jigsaw
3 – Online Jigsaw Treatment	3-person groups	Online Jigsaw

Since the number of members in a main group depends on the number of subtasks and the number of students in the lab section, Table 2 documents the numbers associated with the size of each lab section, the number of main groups, and the number of members in a focus group for the three selected lab topics. Each of the treatment labs had 12 students.

Table 2. Size of main and focus groups for different lab topics.

Lab Topic	**#subtasks**	**|main|**	**|focus|**
Debugging & Testing	4	3 (4 members each)	4 (3 members each)
UML	2	6 (2 members each)	2 (6 members each)
Recursion	3	4 (3 members each)	3 (4 member each)

For each weekly lab, each student was required to take a post-test at the end of the lab individually even when they had been working in a group. However, students working in a group were graded together as a group when their activity worksheets were graded so that everybody in a cooperative group (or main group) was given the same worksheet score. The cooperative students were also asked to evaluate their group members after each selected lab. Therefore, the students had motivations to work together.

5 RESULTS AND OBSERVATIONS

5.1 Jigsaw vs. Individual

To investigate how the Jigsaw model impacts student learning, we compute the average post-test scores for the three selected labs, and compared them to the average post-test scores of the other 11 labs. Table 3 shows the results. Table 4 shows the standard deviations of the post-test scores for the labs.

Table 3. Average post-test scores of the selected labs and the other 11 labs.

Lab Section	#students	11 Labs	Selected Labs	Normalized (Selected/11)
1	17	8.06	7.45	0.92
2	12	7.80	7.47	0.96
3	12	7.12	7.89	1.11

As we can see from Table 3, the online Jigsaw treatment group (Lab Section 3) yielded the best normalized score with 1.11 while the other two groups yielded close to 1.0. We see that the Jigsaw procedure yielded only marginal improvement over the individual lab. This gives some evidence that the Jigsaw procedure could improve student performance.

Table 4. Standard deviation values of the post-test scores of the selected labs and the other 11 labs.

Lab Section	#students	11 Labs	Selected Labs	Normalized (Selected/11)
1	17	1.03	2.37	2.30
2	12	1.29	0.67	0.52
3	12	1.94	1.17	0.61

From Table 4, we see that both Jigsaw treatment groups performed better than the individual control group in terms of the normalized standard deviation values of the post-test scores. That means that the students in these treatment groups were closer in the end in their comprehension of the lab topics causing them to score more similar points on their post-tests. Combining Tables 3 and 4, we see that the in-person Jigsaw treatment group out- performed the individual group in terms of improvement in both post-test scores and the range of post-test scores.

5.2 In-Person Jigsaw vs. Online Jigsaw

In this investigation, we compare the in-person Jigsaw results with the online Jigsaw results. The objective here is to see how well online communication could support the Jigsaw procedure. Referring back to Tables 3 and 4, we observe that students in the online Jigsaw treatment group performed better than students in the in-person Jigsaw treatment group.

The above observation had been unexpected because of two reasons: (1) the Jigsaw procedure is not as effective without face-to-face, free-form discussions, and (2) three key drawbacks of the modified I-MINDS system. The drawbacks are as follows. First, students were more adept at verbal conversation than at typing messages on a chatroom or drawing using a mouse on a digital whiteboard. Second, collaborative programming activities require students to work on and to refer to the same portion of code. Though I-MINDS allowed students to send program snippets to each other, it was not conducive to students to jointly inspect a piece of code. During the in-person Jigsaw group, students enjoyed animated discussions together and drew and wrote on papers to help illustrate their ideas. Third, the students used an Interactive Design Environment (IDE) during each lab. The IDE interface and the I-MINDS interface both occupy entire computer screen, making it inconvenient to use both tools at the same time.

Upon further analysis, we realize that students using the I-MINDS system had to type and send messages to communicate. We suspect that this act of typing caused the students to cognitively process their questions and answers during their interactions with their group members more consciously. Some students working the face-to-face Jigsaw lab could be in a discussion group, listening to the discussion without comprehending. However, students working in the online Jigsaw lab definitely had to communicate to find answers to questions that they could not solve. Further, each student working in the online Jigsaw lab were also able to review all questions and answers communicated within his or her group since all chatroom messages were recorded and displayed in sequence.

5.3 Jigsaw and Laboratory Topics

In this analysis, we probe further how Jigsaw impacted the three different lab topics: debugging and testing, UML, and Recursion. Table 5 shows the results. From Table 5, we see that in-person Jigsaw treatment group did poorly in both the Debugging & Testing and the UML topics, while performed close to the online Jigsaw treatment group in Recursion. The online Jigsaw treatment group performed almost at the same level as the individual control group in both the Debugging & Testing and the UML topics.

Table 5. Normalized average post-test scores for each selected lab topic over the other 11 labs.

Lab Section	#students	Debugging & Testing	UML	Recursion
1	17	0.94	1.16	0.69
2	12	0.89	0.81	1.18
3	12	0.94	1.14	1.25

Referring to Section 3.1 where we discussed the laboratory topics and activities, we draw the following implications. First, it seems that the Jigsaw procedure was helpful to students in understanding the Recursion concept, a lab that has three very distinctive subtasks (or activities). Second, the impact of Jigsaw is minimal when the subtasks to be accomplished require mostly trial-and-error-type of activities and when the subtasks are similar. This is based on the results observed from the Debugging and Testing lab. (Of course, it is also possible that lack of familiarity about Jigsaw when it was first implemented in the Debugging and Testing lab could have pushed the students towards working more individually as well.) Third, the in-person Jigsaw seemed to have failed in the UML lab, with a significantly lower score of 0.81. We speculate that this was due to the size of the focus group. As shown in Table 2, the size of a focus group in this lab was 6. With 6 members in the discussion, it is possible that the focused exploration got distracted in which members followed different threads of discussions simultaneously. It is also possible that due to the configuration of the computer desks in the lab (the students could not sit in a round table fashion), students did not enjoy true in-person interaction. On the other hand, with the online Jigsaw, all messages were tracked, displayed, and students who cared to followed the discussion, would be able to view and review each message. In a way, the online Jigsaw setup seemed to have enhanced the environment for collaboration.

5.4 Other Observations

Our study was not set up to compare the Jigsaw procedure and unstructured cooperative learning. Will students working in the Jigsaw procedure out-perform students working in a group of three with no specified structure? In Spring 2005, students in Lab Sections 2 and 3 worked in the same group of three for those 11 labs but in different groups for each of the selected labs. That is, during those 11 weeks, the students were more likely to develop comfortable and even effective working relationship in their groups to solve programming problems. On the other hand, students in the Jigsaw groups had to work with different main and focus groups. Therefore, if we were to conduct a study to compare the Jigsaw procedure and unstructured cooperative learning, we believe that it would be important to also change the 3-member groups every week in the unstructured cooperative learning labs.

6 CONCLUSIONS

We have applied the Jigsaw cooperative learning model to our CS1 closed labs. Our study investigated the impact of Jigsaw in student learning, the performance of an online Jigsaw design in comparison to an in-person design, and the impact of Jigsaw in three different lab topics. We have drawn interesting observations that seem to indicate that the Jigsaw model can improve student performance and also produce more consistent student performance, and that an online Jigsaw system can have unexpected benefits. Further experiments and additional data collection are definitely needed to obtain substantial evidence to support the above observations.

ACKNOWLEDGEMENT

The ILMDA project was partially supported by an NSF SBIR grant, DMI-044129. The author thanks Nobel Khandaker for helping carry out the above study, Nobel and Xuli Liu for their programming work on I-MINDS, and Hong Jiang for his contribution to the I-MINDS project.

REFERENCES

[1] Aronson, E., N. Blaney, J. Sikes, C. Stephan, and M. Snapp (1978). *The Jigsaw Classroom*, Beverly Hills, CA: Sage.

[2] Bargh, J. A. and Y. Schul (1980). On the Cognitive Benefits of Teaching, *Journal of Educational Psychology*, **72**:593-604.

[3] Clarke, J. (1994). Pieces of the Puzzle: The Jigsaw Method. In S. Sharan (ed.) Handbook of Cooperative Learning Methods, Westport, CT: Greenwood Press.

[4] Cockburn, A. and L. Williams (2001). The Costs and Benefits of Pair Programming, in G. Succi and M. Marchesi (eds.) *Extreme Programming Examined*, Boston, MA: Addison-Wesley Longman, pp. 223-243.

[5] Goodlad, S. and B. Hirst (1989). *Peer Tutoring: A Guide to Learning by Teaching*, Nichols Pub.

[6] Nagappan, N., L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, and S. Balik (2003). Improving the CS1 Experience with Pair Programming, in *Proc. SIGCSE'2003*, February 19-23, Reno, NV, pp. 359-362.

[7] Johnson, D. W., R. T. Johnson, and K. A. Smith (1991). Cooperative Learning: Increasing College Faculty Instructional Productivity, *ASHE-ERIC Higher Education Report No.* 4, George Washington University.

[8] Johnson, D. W., R. T. Johnson, and M. B. Stanne (2000). Cooperative Learning Methods: A Meta-Analysis, Essays by the Cooperative Learning Center at the University of Minnesota, May, http://www.co-operation.org/#essays

[9] Ladd, B. and E. Harcourt (2005). Student Competitions and Bots in an Introductory Programming Course, *Journal of Computing Sciences in Colleges*, **20**(5):274-284.

[10] Liu, X., X. Zhang, L.-K. Soh, J. Al-Jaroodi, and H. Jiang (2003). A Distributed, Multiagent Infrastructure for Real-Time, Virtual Classrooms, *Proc. ICCE'2003*, Hong Kong, China, December 2-5, pp. 640-647.

[11] McDowell, C., L. Werner, H. Butlock, and J. Fernald (2002). The Effects of Pair-Programming on Performance in an Introductory Programming Course, *Proc. SIGCSE'2002*, February 27 – March 3, Covington, KY, pp. 38-42.

[12] Natvig, L. and S. Line (2004). Age of Computers—Game-Based Teaching of Computer Fundamentals, *Proc. ITiCSE'2004*, June 28-30, Leeds, UK, pp. 107-111.

[13] Ploetzner, R., P. Dillenbourg, M. Preier, D. Traum (1999). Learning by Explaining to Oneself and Others, in P. Dillenbourg (ed.) *Collaborative Learning: Cognitive and Computational Approaches*, Oxford: Elsevier, pp. 103-121.

[14] Ravenscroft, S. P., F. A. Buckless, and T. Hassall (1999). Cooperative Learning – A Literature Guide, *Accounting Education*, **8**(2):163-176.

[15] Soh, L.-K. (2004). Using Game Days to Teach a Multiagent Systems Class, *Proc. SIGCSE'2004*, March 3-7, Norfolk, VA, pp. 219-223.

[16] Soh, L.-K., H. Jiang, and C. Ansorge (2004). Agent-Based Cooperative Learning: A Proof-of-Concept Experiment, *Proc. SIGCSE'2004*, March 3-7, Norfolk, VA, pp. 368-372.

[17] Soh, L.-K., A. Samal, S. Person, G. Nugent, J. Lang (2005). Closed Laboratories with Embedded Instructional Research Design for CS1, *Proc. SIGCSE'2005*, St. Louis, MO, February 23-27, pp. 297-301.

[18] Soh, L.-K., N. Khandaker, X. Liu, and H. Jiang (2005). Computer-Supported Structured Cooperative Learning, accepted to *ICCE'2005*, November 28 – December 2, Singapore.

[19] Williams, L., R. R. Kessler, W. Cunningham, and R. Jeffries (2000). Strengthening the Case for Pair Programming, *IEEE Software*, **17**(4).