ACTIVE LEARNING WITH UPPER DIVISION COMPUTER SCIENCE STUDENTS

Brenda Timmerman[1], Robert Lingard[2] and G. Michael Barnes[3]

Abstract *— Computer Science students especially upper division students are stereotypically considered to be introverted and therefore poor candidates for an Active Learning curriculum. Ironically, the requirements of their field demand skills in critical analysis and evaluation, as well as communication and collaboration skills, that are not easily acquired in the traditional classroom environment with a "lecture" agenda where students maintain a basically passive role. This paper describes experiments with Active Learning techniques performed in two different upper division Computer Science classes, "Societal Issues in Computing" and "Computer System Security". In spite of the traditional views of Computer Science as an individualistic subject matter with self-centered non-social students, the authors have used Active Learning techniques in their classrooms for several semesters with encouraging results. In addition to an improved attitude and stated increased satisfaction, students' test results showed increased comprehension and improved critical reasoning abilities.*

Index Terms — Active Learning, communication skills, cooperative learning, teamwork.

INTRODUCTION

Active learning in the classroom includes nearly every activity other than merely passively listening to an instructor's lecture. Short writing exercises, sharing information in student pairs or groups, and complex group problem solving exercises are all examples of active learning. Cooperative learning is that subset of active learning that usually involves formally structured groups of three or more students assigned multi-step exercises, research or development projects, or presentations. According to social psychological theories, learning is more effective when process is an active rather than a passive one [1]. Many studies have been done that support this notion. Ruhl, Hughes, and Schloss demonstrated the dramatic improvement on long term retention of course information by merely inserting two minute pauses after every fifteen minutes or so of lecture to allow students to work in pairs to discuss and revise their notes [2]. In a more directly relevant result, McConnell has shown a statistically significant correlation between the use of active learning exercises and final exam scores for students in a theory of computation course for computer science majors [3].

It is clear that active learning works, but there are additional reasons for utilizing certain communication and group oriented learning techniques that are characteristic of active learning within engineering and computer science courses. Both educators and industry representatives seem to agree that to be successful in today's workplace high levels of teamwork and communication skills are needed by engineering and computer science graduates [4]. Feedback from employers of our graduates indicates weakness in both written and oral communication skills and inexperience in working in groups. Furthermore, ABET (Accreditation Board for Engineering and Technology) has emphasized both communication and teamwork skills in the 2000-2001 criteria for accrediting engineering programs. Specifically, the guidelines state that "engineering programs must demonstrate that their graduates have . . . an ability to function on multi-disciplinary teams [and] . . . an ability to communicate effectively" [5].

STUDENT CHARACTERISTICS

With all these reasons for using active learning techniques, why are they so infrequently used, especially in engineering and computer science courses? Aside from the fact that it is difficult to get instructors to change their style of teaching, there seems to be a belief that most forms of active learning are inappropriate in engineering and computer science courses. It is believed that students in these majors don't like to talk and prefer to work alone rather than in groups. There is, in fact, evidence to support these claims.

In a study conducted within the Computer Science department at CSU Northridge, most majors were shown to be introverted. Using the Myers-Briggs personality test [6] on 304, only 37% were classified as extroverted while 47% were introverted and 16% were evenly split. The percentage norms for the general population are 75% extroverted and 25% introverted. The complete results of this study are shown in Table I.

The typical computer science major is an ISTJ Myers-Briggs type. These types take their energy from their inner thoughts. They like to deal with facts and make decisions

[1] Brenda Timmerman, California State University, Northridge, College of Engineering and Computer Science, Northridge, CA 91330-8281, btimmer@ecs.csun.edu
[2] Robert Lingard, California State University, Northridge, College of Engineering and Computer Science, Northridge, CA 91330-8281, rlingard@ecs.csun.edu
[3] G. Michael Barnes, California State University, Northridge, College of Engineering and Computer Science, Northridge, CA 91330-8281, renzo@csun.edu

considering various options. They are quiet, serious, and like to be prepared for what may happen.

TABLE I
MYERS-BRIGGS RESULTS FOR COMPUTER SCIENCE MAJORS

	Computer Science Majors	General Population
Extroversion (E)	37%	75%
Introversion (I)	47%	25%
E/I Split	16%	
Sensing (S)	48%	75%
Intuition (N)	38%	25%
S/N Split	14%	
Thinking (T)	57%	50%
Feeling (F)	35%	50%
T/F Split	8%	
Judgement (J)	78%	50%
Perception (P)	14%	50%
J/P Split	7%	

In another study conducted by one of the authors [7] the Kolbe A™ Index was used to measure conation, or students' natural approach to problem solving. Most software engineering students were found to function most naturally in the Fact Finder mode. According to Kolbe, such individuals are likely to be uncomfortable sharing information before they have all the facts [8]. Certain types of active learning exercises could be stressful for such individuals.

The Kolbe A Index is an instrument that measures conation or a person's inherent talent or natural way of doing things and predicts what a person will or will not do, given the freedom to act. Whereas intelligence tests measure I.Q. and personality tests measure values and preferences, the Kolbe index measures the conative, the way people act while trying to achieve goals. It identifies four modes or striving instincts -- Fact Finder, Follow Thru, Quick Start, and Implementor -- each prompting people to act in a certain way. The Fact Finder collects data and establishes priorities before making a decision. The Follow Thru individual seeks structure and makes schedules. The Quick start individual innovates, takes risks, improvises, and plays hunches. The Implementor uses space and materials, builds, constructs, and uses hands-on equipment with ease. Everyone has each of these abilities to some degree. However, people are most productive when they are able to utilize their strongest conative talents.

The Figure 1 graphically depicts the degree to which each of these abilities is present. The four striving instincts are expressed through three possible operating zones, indicating how the individual will make use these talents. A score of 7 to 10 in a given mode places the individual in the insistence zone. This indicates *how the person will act*. A score of 4 to 6 indicates the response or accommodating zone or *how the person is willing to act*, and a score of 1 to 3 represents the prevention or resistance zone or *how the person won't act.*

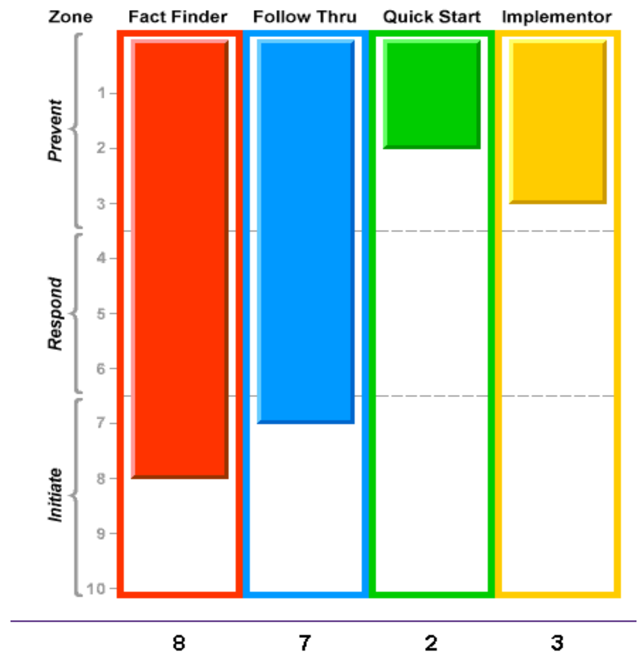

FIGURE 1
SAMPLE KOLBE RESULTS

Although, in general, there is no correlation between the Myers-Briggs results and those of Kolbe [9], our research has shown that computer science students tend to be both introverted in the Myers-Briggs sense and Fact Finder/Follow Thru individuals under the Kolbe analysis. Specifically, of 181 software engineering students who were given the Kolbe A Index, 87 (48%) had Fact Finder as their primary mode of operation, and 116 students (64%) had Fact Finder as their first or second highest mode. This is nearly twice the expected amount based on the general population distribution. Nearly half of the students (90 of the 181 students) had Follow Thru as their primary or secondary mode of operation.

The complete distribution of Kolbe modes for the software engineering population studied is shown in Table II. For the general population the expected results are:

- 20% initiating action in each of the four Kolbe Action Modes
- 60% responding to people and situations through each mode, and
- 20% resistance to taking action in each of the modes.

TABLE II
KOLBE RESULTS FOR SOFTWARE ENGINEERING STUDENTS

	Fact Finder	Follow Thru	Quick Start	Imple-mentor
Resistance	2%	5%	62%	30%
Responding	44%	62%	33%	64%
Initiating	54%	33%	5%	6%

The typical computer science student, in the Kolbe sense, will avoid discussing issues without being prepared ahead of time and will prefer to explain things with charts and diagrams rather than words. Impromptu discussions will create stress for such individuals, and they will resist participation [10].

It is interesting to note that these characteristics of computer science students are also typical of the computer science faculty. The Kolbe A Index was given to several members of the faculty and the results showed an even stronger fact finder component. This suggests that a possible reason why computer science faculty are reluctant to use some active learning techniques is because they also feel uncomfortable learning in this mode.

Other characteristics of CSU Northridge that would seem to present challenges for active learning are cultural diversity within the student population and the large number of students with disabilities, especially hearing impairments. In one software engineering class of 32 students there were 14 different languages specified as the student's native language on an end of semester survey. Typically, the ratio between students and native languages is less than three to one. Fewer than 35% students specified English as their native language. Since English is the second language for nearly two thirds of most classes there can be some inherent communication problems in group discussions and projects. Although students with disabilities are not present in such large numbers, there are typically one or two such students in each class. Students who are deaf or hearing impaired are the most common although blind students and students with other physical or learning disabilities are also often present. CSU Northridge is deeply committed to meet the educational needs of students with disabilities. Through its National Center on Deafness, CSU Northridge has become recognized as a national leader in providing quality education to the deaf and hard of hearing. It was the first mainstream university in the nation to provide full-time professional interpreters in the classroom.

Despite these challenges, however, efforts to use active learning in upper division computer science classes have been very successful. Some care has been taken to choose exercises that are well suited to the learning styles that are typical among engineering and computer science students and are sensitive to the cultural diversity and disabilities present within the classroom. The following sections describe the specific forms of active learning used and present an assessment of the results. Some conclusions based on these experimental efforts in active learning are drawn and recommendations for future work are made.

EXPERIMENTS WITH ACTIVE LEARNING

One of the computer science classes for which it seemed most natural to employ active learning techniques was the course on "Societal Issues in Computing". This course is a senior level course required of computer science majors. This course examines ethical issues such as privacy, copyright, and free speech as well as the impact of computers on education, productivity, and health and safety. Group discussions, debates, and team presentations seem to be natural ways to promote learning of the material. Some early attempts, however, met with difficulties. For example, when THINK-PAIR-SHARE was used as a learning mechanism, the results were mixed. In THINK-PAIR-SHARE students first write down their thoughts, then they talk about them with one other person, and finally share them in a group which presents the consensus response to the question or problem to the rest of the class. A typical question might be "Why are there so few women and minorities majoring in computer science?"

Although appropriate reading assignments were given, if the specific discussion questions were not known until class time, many students were not adequately prepared to participate in the discussion. Students accustomed to the traditional lecture course often feel that they need not do the assigned reading ahead of time, if at all, since they will probably be able to get what they need to know from the instructor's presentation. Students who are, conatively speaking, fact-finders have difficulty participating in such discussions if they are not prepared. Many such students will just sit and listen to the rest of the group, reluctant to participate. To maximize the effectiveness of this technique for computer science students, it is advisable to announce the specific questions that will be discussed in advance. In that way the fact-finders will be able to prepare and be more likely to participate in group discussions. An added benefit of this approach is that more students actually do the assigned reading before the material is covered in class.

Another active learning technique used in this class was GROUP PRESENTATIONS where a group of students must research a topic and prepare a presentation for the rest of the class. The topic to be discussed might be the issue of "privacy of personal information". The presentation would be organized by the group, but each panelist would be expected to make a short presentation. The floor is then opened to questions from the audience (the rest of the class). This exercise works fairly well for computer science students since the group is given adequate time to do research (fact find) before making their presentation although the thought of questions being asked can be stressful. A variation of this technique, DEBATES, was also used. In this case there are two groups and each presents a different side to an issue, like "Should there be laws to control pornography on the Internet?" The debate can be controlled by letting each group present arguments in support of their position and then letting each rebut the arguments of the other. If time permits each group can then respond to the rebuttals.

Another computer science class that seemed like it would benefit from the use of active learning techniques was "Computer System Security". This is a senior and graduate level course and is part of the cores of the Computer Network and Operating Systems Concentrated Studies Packages. By their Junior year all Computer Science majors at CSU Northridge must choose such a package that is appropriate to their career objectives. Computer Security is a relatively new field that can be quite controversal. Any study or implementation of computer security requires a good deal of critical thinking and evaluation on the part of students.

Currently computer security and ethics are of great interest to the general population. Daily articles and other items on these subjects appear in all forms of news media. This made it very easy to present the pertinent issues of each of the courses in their historical and real world context by starting each class with a discussion of current events in computer security or societal issues involving computers. Students were encouraged to bring in articles and share them with the class. Some of the more introverted students participated particularly well in these sessions. Possibly this was because they had time to prepare in advance and felt more confident. With the articles in their hands, they were the experts on the issue being discussed.

As part of cooperative learning the students were divided into groups in order to discuss the issues raised during the 15 or 20 minute lectures that preceded the group sessions. Some important essentials for successful groups that the professor has to be aware of are: appropriate grouping, individual accountability, professor as facilitator, and an end product. Each group session has to end with the group producing something such as a report on the conclusions of the group discussion [11].

A technique called KEY WORDS was used during cooperative learning sessions where the students separated into groups of four or five for discussions and reports to the class on the groups' conclusions made from the discussion at the end of the session. Group discussion questions that were assigned during the sessions contained key words recommended by Moss and Holder [12] such as *evaluate, contrast, explain, describe, define, compare, discuss, criticize, prove,* and *illustrate.* These words were chosen to help them develop their critical thinking skills as well as their writing skills. These same key words were used on exams to give them additional practice and to assess the improvement in their writing skills. Another technique that was used was to have the groups discuss and report on news items brought to class and have the students keep journals of these items. The students knew that the journals were to be collected at the end of semester.

An active learning technique that was found to be very valuable was the ENTRANCE and EXIT SURVEYS. These had many applications. They could be used by the teacher for ongoing assessment as discussed below or they could be a powerful tool to help the student become involved in the learning process. An important feature of the SURVEY method is that the students receive feedback on the results of the surveys. A survey could ask questions about the course material or ask the student to evaluate the class or the instructor. It was observed that the students took these surveys very seriously and were very attentive when the results of the surveys were reported back to them.

ASSESSMENT

We were initially concerned about how to assess our efforts but found that many accepted assessment techniques were appropriate for our experiments. Angelo and Cross describe fifty Classroom Assessment Techniques (CATs) that have been proven to be useful in class assessment [13]. These techniques have been successfully adopted by faculty throughout the United States and are well recognized for their value for individual courses. Their use in program assessment is less well documented.

Simple techniques were chosen for initial and ongoing assessment. These include The Minute Paper, The Muddiest Point, and The One-Sentence Summary. These classroom strategies were used to improve learning because they helped us to evaluate the classes progress towards our defined objectives, and they let us know if we needed to change directions. The Minute Paper was assigned to answer the following questions: What was the most important thing you learned today? What questions remain uppermost in your mind as we conclude this session?

The Muddiest Point simply asks the question: What was the muddiest point in today's class session? This was best done as an EXIT SURVEY. Exit surveys were good indicators of students' satisfaction and their current opinions. They were used to track opinions over the semester. M. J. Allen advises the use of simple clear cut questions in an Exit Survey [14]. We followed his advice and got the best results when we avoided compound questions, vague questions, or confusing or biased questions. The One-Sentence Summary was used during a break in the lecture. A sample question might be in the following form: Encryption might be installed in a computer system by whom, in order to mitigate what, by whom, where, how, and why?

The midterm and final exams in the Computer System Security classes used a method of evaluation suggested by Jacobs and Chase [15]. Essay questions containing the KEY WORDS mentioned above were used in the exams. Some examples of exam questions are: "Contrast the substitution method for data encryption with the transposition method."; "Present arguments for and against mandatory access control."; and "What are the relevant parts of system security auditing?" The control testing was done with students of Computer System Security who were taught with the lecture method. We compared the tests results of these students with the results of the students in classes that were exposed to the active learning techniques described above. Though all classes were given the same or very similar tests, the students exposed to active learning gave answers that were more correct, more comprehensive, and an average of 50% longer, showing that the students were not only more confident, but felt they had more to say about the subject. One weakness with this method is that even though it was recommended and was useful to us, its reliability and validity is generally unknown.

Another assessment technique that we used was the observation method suggested by Babbie [16]. Observation is considered to be able to provide more valid data about social processes than some other data collection strategies. The classes were evaluated by observing the students interacting during their group sessions. In this case, the observers, the professors, were usually not part of the social process that was being observed. However, the professors frequently had to answer questions and act as facilitators for the groups, and this is acceptable for the observer in the method. An improvement was noted in the students' ability to interact as the semester progressed. The method allowed us to observe the subtle nuances of attitude and behavior that are difficult to measure quantitatively.

Of particular interest was the observation of the hearing-impaired students who made up about 5% of the groups. Though communicating with the other students with the assistance of an interpreter, either signing or using a closed-caption monitor, was awkward at first for all involved, the hearing-impaired students, the rest of the groups, and their interpreters were observed to quickly adjust and become comfortable groups. The hearing-impaired students sometimes took leadership roles in the groups, such as presenting the final reports.

SUMMARY AND CONCLUSIONS

The efforts to ulitize active learning techniques in upper division Computer Science courses were extremely successful. Concerns that the nature of the material and the characteristics of the students would be obstacles to success proved to be unfounded. It was not difficult to create active learning exercises for the classroom that reinforced the concepts of the courses, and the students generally participated with enthusiasm.

As observed in this study, students in classes where active learning was practiced had improved attendance, demonstrated improve communication skills, achieved higher test scores, and generally appeared more interested in the course material. We attribute this success, in part, due to the care that was taken to design cooperative exercises that were appropriate for the type of students taking upper division computer science courses. In particular, students were usually given adequate time to prepare for the active learning exercises thereby reducing the stress that might be associated with an impromptu activity. The excerises were well defined in an effort to satisfy the students' needs for organization and structure. Active learning can be used successfully in all fields, but it is important to consider the nature of the student in order to design effective classroom exercises.

REFERENCES

[1] Sherman, Lawrence W., "Cooperative Learning in Post Secondary Education: Implications from Social Psychology for Active Learning Experiences", http://miavx1.muohio.edu/~lwsherman/aera906.html, 1991.

[2] Ruhl, K. L., Hughes, C. A., and Schloss, P.J., "Using the Pause Procedure to Enhance Lecture Recall", *Teacher Education and Special Education,* Vol. 10, 1987, pp. 14-18.-

[3] McConnell, Jeffrey J., "Active Learning and its Use in Computer Science", *SIGCSE Bulletin,* Vol. 28, Special Issue 1996, pp. 52-54.

[4] Lewis, Philip, Aldridge, Dayne, and Swamidass, Paul M., "Assessing Teaming Skills Acquisition on Undergraduate Project Teams", *Journal of Engineering Education,* Vol. 87, No. 2, 1998, pp. 149-155.

[5] Accreditation Board for Engineering and Technology, Inc., "Criteria for Accrediting Engineering Programs, http://www.abet.org/eac/eac.htm, 1999.

[6] Briggs-Myers, Isabel and McCaulley, Mary H., *Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator,* Consulting Psychologists Press, 1985.

[7] Lingard, Robert and Berry, Elizabeth, "Teaching Communication and Teamwork in Engineering and Computer Science", *Proc. ASEE Annual Conference & Exposition,* 2001.

[8] Kolbe, Kathy, *The Conative Connection,* Addison-Wesley, Reading, MA, 1990.

[9] Kolbe Corporation, "Statistical Analysis of Kolbe Indexes", Kolbe Corp., 2000.

[10] Kolbe, Kathy, *Pure Instinct,* Times Books, New York, 1993.

[11] Cooper, J. and Mueck, R., "Student Involvement in Learning: Cooperative Learning and College Instruction" *Journal on Excellence in College Teaching*, 1990, pp. 68-76.

[12] Moss, A. and Holder C., "Improving Student Writing", Kendall and Hunt, Dubuque, IO, 1988.

[13] Angelo, T. A., and Cross, K. P., "Classroom Assessment Techniques: A Handbook for College Teachers" (2nd ed.), Jossey-Bass, San Francisco, 1993.

[14] Allen, M.J., "Introduction To Psychological Research", Peacock, Itasca, IL, 1995.

[15] Jacobs, L.C. and Chase, C. I., "Developing and Using Tests Effectively: A Guide For Faculty", Jossey-Baas, San Francisco, CA, 1992.

[16] Babbie, E., "The Practice of Social Research", Wadsworth, Belmont, CA, 1995.