

Active Learning for Classroom Management Model

Keith Whittington and Michael Yacci
**Golisano College of Computing and Information Sciences,
Rochester Institute of Technology, Rochester, NY, USA**

kjw@it.rit.edu; may@it.rit.edu

Abstract

This paper proposes the Active Learning for Classroom Management Model as a method to structure active learning experiences. The model was devised and tested in an Introductory Computer Programming course. Stages in the model are explained with examples of classroom activities. The model was evaluated twice in series of quasi-experiments and limited support was provided for improved student performance through the use of the model.

Keywords: Active Learning, Learning Model, Innovation, Programming

Introduction

Approximately three years ago, courses were developed for at-risk students in introductory computer programming courses in a large college in upstate New York. A suite of teaching techniques, centered around *active learning*, were chosen to combat the problem, resulting in a grant from The National Science Foundation (NSF) to support and study this work. The main goals of the grant were to show evidence of student learning and satisfaction, develop discipline specific active learning activities, and to provide workshops. During the two years of evaluation and implementation of the grant, the basic design of the courses emerged as a more general learning model. In this paper we propose the Active Learning for Classroom Management Model (AL-CM) as a plausible account for the improvement in student learning and present evidence to support the model.

Background

Introductory Computer Programming, like many difficult technical subjects, faces a large number of student failures and withdrawals. Some of the failures and withdrawals are due to the ways in which these courses are commonly taught in most universities using traditional teaching methods of lecture, individual assignments and projects. These methods themselves are not to blame, but

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

rather it is the overall combination of these methods that produces a passive classroom atmosphere that promotes memorization of material rather than understanding. Students struggle on their own to master material that is confusing. To help alleviate these classroom problems, active learning techniques were developed for trial sections of Introduction to Computer Programming courses. In particular, *cooperative learn-*

ing was chosen as the primary method for the active learning classroom activities. This was primarily due to its ability to create genuine communities within classrooms, and deepen learning via well-structured, sequenced assignments in small groups (Millis & Cottell, 1998). It also gives students the opportunity to talk, listen, read, write, reflect, and apply what they are learning (Meyers & Jones, 1993). Cooperative learning seemed promising for younger students attending college for the first time, who may be a little apprehensive and unsure of their abilities.

One of the keys in the application of active learning to the course design was the notion that portions of a course should support each other to create a significant learning experience (Fink, 2003). Additionally, several classroom management techniques and activity designs were studied and adopted from Millis & Cottell (1998).

It should be noted that the major motivator for these design activities was to improve classroom practice by achieving higher student achievement, fewer failures, and higher student satisfaction. The original intent was *not* to develop a generalized classroom model. However, as is typical in grounded theory situations (Glaser & Strauss, 1967), a model gradually emerged and solidified itself through the process of data gathering and analysis.

Developing an Active Learning Model

Because the AL-CM is an emergent model, based on successful practice, it is appropriate to describe the process that led to the development of the proposed Active Learning for Classroom Management (AL-CM) model.

Studio Model

The course, Introduction to Computer Programming, as it initially existed, was a *studio model* design in which the lab portion of the class happened immediately after a short lecture, in the same room, and with the same instructor. Lab activities are generally short, simple applications of the lecture material. This was a powerful way for instructors to receive immediate feedback on the effectiveness of the lecture, and discover any students' lack of understanding. Despite this seemingly active approach, students still worked individually, and the class time was dominated by instructor lecture.

Building a Toolbox of Activities

As a pilot project, active learning activities were created and used in a trial classroom as the need arose. When a general lack of student understanding on a particular concept occurred, for example, an activity was created to address this need. Several times during the courses, students rated the effectiveness of each activity and were encouraged to make suggestions for improvement. The activities were modified based on this feedback and used in subsequent versions of the courses. This critical reflection by the students was paramount to the ultimate effectiveness of all the activities. Brookfield (1995) discusses several powerful ways to get students to critically reflect on their learning and to inform the instructor's practice.

A cooperative approach was used for the in-class activities where paired programming was employed for the in-class activities where two programmers work on one program. In a paired programming activity, one student assumes the role of the "driver" who writes the code while the other programmer assumes the role the "navigator" who makes suggestions, and corrects coding and logic errors as they occur (Astels, Miller, & Novak, 2002; McDowell, Werner, Bullock, & Fernald, 2002).

Nurturing the Classroom Environment

A more concerted effort was made to make all phases of the learning environment more active and learner-centered as advocated by Weimer (2002). One adaptation occurred with the paired-programming exercise when it was observed that the “navigating” students appeared to lose focus. Therefore, the in-class activities were redesigned such that each step was numbered and the students were required to switch roles for each step. This minor change had a dramatic effect at keeping the pairs on task and generating a high level of student engagement.

The other addition to the classroom was the addition of Think-Pair-Share (TPS) into the lectures. TPS requires that when the instructor poses a question, the students think about the question, pair-up with another student, and share their answers. The benefits of this is to give students time to think about their answers, engages them by discussing the topic, lets them learn from each other, raises their self-confidence by having their knowledge verified, and possibly feeling better that they weren't the only ones who didn't know the answer (Bonwell & Eison, 1991; Sutherland & Bonwell, 1996).

Putting the Pieces Together

In the process of trying to show evidence of student learning, developing discipline specific activities, and assessing student learning and engagement, a learning model started to emerge. It became apparent that actively engaging students could be done in a systematic way. The following model attempts to capture the process that maximizes the effectiveness of infusing active learning into the classroom.

Learning Questions

We propose a four-phased approach for the AL-CM. In particular, we believe that each phase of the model must address specific learning questions that are relevant to students at different times. Figure 1 shows the four major components of this learning model:

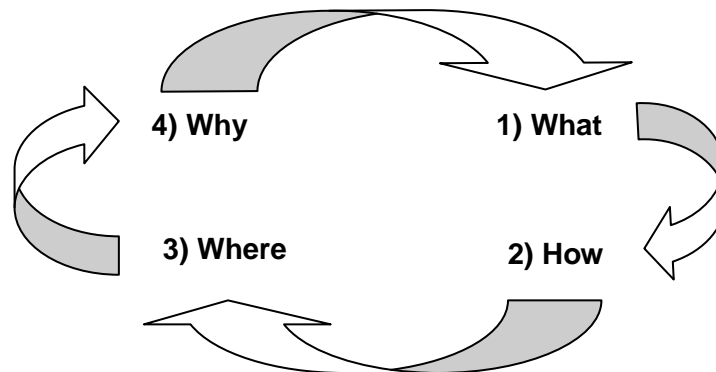


Figure 1: Learning Questions

During the first activity phase, it is important to show the students **what** to do. Students need orientation and direction to get started efficiently. The second phase helps them to learn **how** to do it. That is, they learn the mechanics and procedures for accomplishing small tasks and basic implementations of the ideas. The third phase helps them to combine their new knowledge with existing knowledge to see **where** the pieces fit together conceptually. The final phase helps the student to know **why** they are learning the current material and the purpose behind it.

Figure 2 shows the AL-CM learning questions mapped onto a classroom implementation and the following sections provide a description of the implementation of each phase of the model.

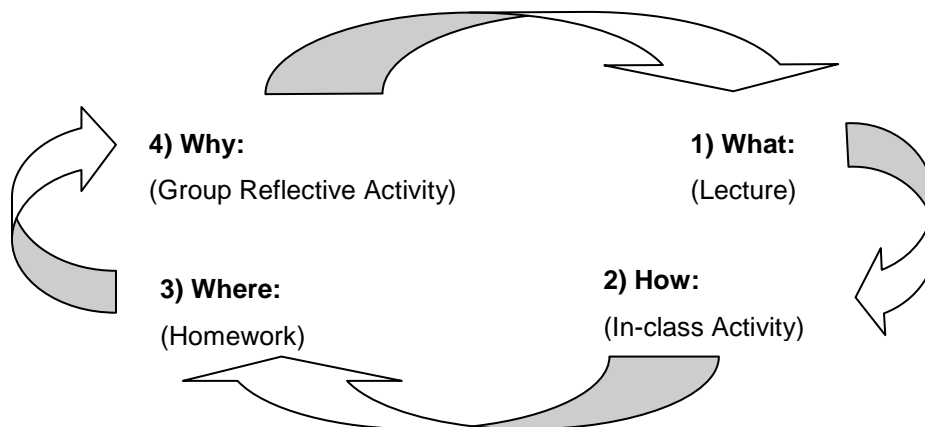


Figure 2: Active Learning Model for Classroom Management

Phase 1 – What: Lecture

Although this is an active learning model, it does not advocate the exclusion of lecturing. Lecturing is still a very efficient way explain new concepts and constructs; a great lecturer can still captivate an audience. However, lecturing is most effective in transmitting information (Bligh, 2000). A one-way transmission was found to be useful for setting the stage for students, for providing an overview of material and for providing general directions and modeling.

Phase 2 – How: In-class Activity

The purpose of an in-class activity is to give the students immediate practice with the recently covered material by solving simple problems. It also allows the students to ask questions while the lecture is fresh in their minds and the instructor is still in the room. It enables students work with the procedural and technical aspects of the new content in a complex, local environment such as a laboratory or virtual space.

The techniques of paired programming were implemented for this part of the learning model. Paired programming is an effective way to keep programmers active in developing programs at both the conceptual level (navigator) and the operational level (driver). Interestingly, after several sessions, student pairs consulted with other pairs before asking the instructor for help. It may be the case that students who figure out procedures for themselves have a perspective on the subject matter that resonates with other beginners and therefore are effective in explaining ideas to each other. Nonetheless, when students did ask the instructor a question, it was typically well thought out and discussed with at least two people.

Phase 3 – Where: Homework

Homework assignments should be a more difficult application of the current topic to allow students to put together all the pieces of the new concepts and create something on their own, which supports deep learning (Zull, 2002). This is also where the students integrate their new knowledge with their existing knowledge; essentially they can see where new ideas fit with existing ideas. This can be an even richer learning experience when the current homework expands on the previous homework. As students experience success in this application, it provides an incentive to learn the next topic. The suggestion is that students should do these assignments on their own, although getting help from other students and instructors is generally accepted.

Phase 4 – Why: Reflective Activity

This is the critical phase of this learning model and where students reflect on their current knowledge to deepen their understanding and expand their existing conceptual models (Zull, 2002).

This phase should use cooperative learning activities where small groups solve a problem or answer questions that are too difficult for most students to solve alone. However, the activities shouldn't be so difficult that the groups give up or too easy where they think it's a waste of time. Examples of specific activities used in these courses are elsewhere described in detail (Whittington, 2004).

One of the problems with cooperative learning exercises is the amount of time they take (Meyers & Jones, 2003), which is why these activities are done in the final phase. In order to maximize the effectiveness in a minimal amount of time, it is suggested to implement these types of activities after the students have a greater level of experience with the topic. Students can begin to build a larger context for why the material they are learning is valuable; these activities can be an effective bridge leading into the need for the next topic.

Implementation of the Model

This model does not require that each part is only performed once as the diagram may imply. Each phase can be done several times. For instance, the following order could effectively be implemented for a particular topic: lecture, in-class, lecture, in-class, homework, lecture, in-class, reflective activity. The crucial points here are to include all phases for each topic or concept, have each part support and build on each other, and ultimately lead into the next topic.

A circular and cyclical model could actually begin at any of the phases, but we suggest starting by providing orientation and direction as suggested in the "How" Phase 1. A study investigating learner preferences (Yacci, 1994) showed several functional learning patterns that commonly arise if students are given choices of learning activities. The AL-CM model and its flexibility is congruent with these learning patterns.

Support for the Model

As stated earlier, the original intent of all of these activities was not to create a general purpose design model, but rather to enhance learning in a specific classroom setting. Data on student achievement, however, provides limited support for this design model.

The Active Learning for Classroom Management model was tested against the traditional studio model in a quasi-experiment (Cook & Campbell, 1979) using two existing Introduction to Computer Programming classrooms and a convenience sample of 34 total students. Students were initially self-selected in the *Active Learning* (N=22) or *Control* classrooms (N=12). The Active Learning treatment group received the instruction designed according to the AL-CM model while the Control group received traditional studio classroom methods. The study occurred over a two week period and both groups were relatively isolated from each other.

Student achievement (pre-test and post-test) was measured on a seven-item paper-based exam, that required each student to analyze sections of computer code and to explain what was happening. This is a commonly-used examination method of assessing student learning in computer programming courses. A unit of instruction was selected that had a clear starting and stopping point, with no carryover from past courses. A two group pre-test / post-test study was conducted to determine if there were differences in achievement on that unit. The same instrument was used as both the pre-test and post-test with minor modification as noted below.

Each group received the same pre-test at the start of an instructional unit. Both groups then received the identical number of classroom meetings, the same total instructional time, and classes

were held in the same computer lab-based facilities. The major difference between the two groups was the inclusion of the AL-CM activities. The control group instructor was told to teach the class as he normally would, using his best teaching methods. At the completion of the unit, after approximately two weeks, both groups received the same post-test. The questions on both pre-test and post-test covered the same concepts and principles, with slight variation in wording and examples.

Random assignment is commonly used to create equivalent treatment groups in experiments. Because there was no random assignment in this experiment, the pre-test was used to show that the two groups did not differ in programming ability as they began the unit. Pre-test scores between the Active Learning and Control group were compared using an independent sample t-Test. Due to the fact that the standard deviations of both groups were similar, the pooled variance was used (pooled SD = 1.5954). Additionally, note that participants were occasionally absent during the administration of the pre-test or post-tests, hence, the varying numbers within each group.

In all experiments, an $\alpha = .10$ was selected as a common (Devore, 2004, p. 323), but not overly rigorous threshold of statistical significance. Triola (2006) alternatively suggests simply reporting the P -value obtained and leaving the decision of statistical significance to the reader. We have done both, in reporting the following results.

	N	Mean	StDev	SE Mean
Active	22	2.45	1.68	0.36
Control	10	3.10	1.37	0.43
T-Value = -1.06 P-Value = 0.297 DF = 30				

As shown in Table 1, using an $\alpha = .10$, the obtained $P = .297$ is not significant, therefore a null hypothesis of equal means ($\mu_0 = \mu_A$) cannot be rejected in based on this data. We therefore retain the null hypothesis of equal means, noting that the logic of hypothesis testing does not prove the null hypothesis, but rather fails to disprove it. This statistic, however, suggests that the two group means are not different on the pre-test.

A similar t-Test for independent means was conducted to compare post-test results. A null hypothesis of equal means was tested. The assumption of equal variances was again used in the computing the two-tailed statistic. Note that, as shown in Table 2, there is one absence from the active learning treatment group.

	N	Mean	StDev	SE Mean
Active	21	6.905	0.889	0.19
Control	12	6.25	1.14	0.33
T-Value = 1.84 P-Value = 0.076 DF = 31				

The Active Learning treatment group scored higher than the Control group. Using an $\alpha = .10$, these data are statistically significant, with a P -value = .076.

These results were replicated one year later in a similar experiment with similar results. Again, self-selected groups were compared, using the same pre and post-tests. In the interests of preserving space, the details of this replication are similar in all aspects to the original experiment, except for the group size. The pre-test comparison is reported in Table 3.

	N	Mean	StDev	SE Mean
Active	34	2.68	1.61	0.28
Control	17	2.41	1.46	0.35

T-Value = 0.57 P-Value = 0.571 DF = 49

Similar to the first experiment, we fail to reject the hypothesis of equal means, with a P -value = .571. The entering skills of the replication groups are not significantly different.

The post-test comparisons are shown in Table 4.

	N	Mean	StDev	SE Mean
Active	34	6.97	1.19	0.20
Control	17	6.41	1.00	0.24

T-Value = 1.66 P-Value = 0.104 DF = 49

On the replicated post-test comparison, the P -value = .104 is not significant at an $\alpha = .10$. However, similar to the first experiment, the Active treatment group once again scored higher than the Control treatment group. Salkind (2008) suggests the term *marginally significant* for data that approaches but slightly exceeds an alpha cut-off level, explaining that significance is not the “end-all” of scientific research. In this situation, in which data approach significance, as an aid to interpretation we provide the P -value. The replication study again suggests that there is practical potential in using the AL-CM model.

Conclusion

The results of these two studies provide limited support on the claim that the Active Learning Classroom Management Model is an effective means to increase student achievement. The original study shows statistical significance in post-test achievement scores; its replication provided marginally significant differences in post-test achievement scores. Results might be due to the length of the experimental conditions; a two-week window may not be enough time to fully and cleanly separate the two treatment groups. We suspect that these differences could be increased if the length of the treatment were longer. Notably, in both experiments, the Active Learning group scored higher than the Control group. The replication of the study, and the consequent similarity in results, leads us to continue to believe the AL-CM has potential for being a practical method for improving classroom achievement and an effective method to structure classroom activities.

An additional report is forthcoming to provide support for the salience of the four distinct phases in the model (Yacci & Whittington, 2008).

References

- Astels, D., Miller, G., & Novak, M. (2002). *A practical guide to extreme programming*. Prentice Hall.
- Bligh, D. (2000). *What's the use of lectures?* San Francisco: Jossey-Bass.
- Bonwell, C., & Eison, J. (1991). *Active learning: Creating excitement in the classroom*. ASHE-ERIC Higher Education Report No. 1. Washington, DC
- Bossert, S. T. (1998). Cooperative activities in the classroom. *Review of Educational Research*, 15, 225-250.
- Brookfield, S. (1995). *Becoming a critically reflective teacher*. San Francisco: Jossey-Bass.
- Cook, T., & Campbell, D. (1979). *Quasi-experimentation: Design and analysis issues for field settings*. Boston: Houghton Mifflin.
- Devore, J. L. (2004). *Probability and statistics for engineering and the sciences* (6th ed.). Belmont, CA: Brooks/Cole-Thompson Learning.
- Fink, L. D. (2003). *Creating significant learning experiences*. San Francisco: Jossey-Bass.
- Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory: Strategies for qualitative research*. Chicago, IL: Aldine Atherton.
- McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). The effects of pair-programming on performance in an introductory programming course. *SIGCSE Bulletin*, 34(1), 38-42.
- Meyers, C., & Jones, T. B. (1993). *Promoting active learning: Strategies for the college classroom*. San Francisco: Jossey-Bass.
- Millis, B. J., & Cottell, P. G. J. (1998). *Cooperative learning for higher education faculty*. Westport, CT: Oryx Press.
- Salkind, N. J. (2008). *Statistics for people who (think they) hate statistics* (3rd ed.). Los Angeles: Sage
- Sutherland, T. E., & Bonwell, C. C. (1996). *Using active learning in college classes: A range of options for faculty*. *New Directions for Teaching and Learning*. Jossey-Bass. San Francisco.
- Triola, M. F. (2006). *Elementary statistics* (10th ed.). Boston: Pearson.
- Weimer, M. (2002). *Learner-centered teaching*. San Francisco: Jossey-Bass.
- Whittington, K. J. (2004). Infusing active learning into introductory programming courses. *The Journal of Computing Sciences in Colleges*, 19(5), 249 - 259.
- Yacci, M. (1994). A grounded theory of student choice in information-rich learning environments. *Journal of Educational Multimedia and Hypermedia*, 3(3), 327-350.
- Yacci M. & Whittington, K. (2008). What is the active ingredient in active learning? *Proceedings of the 2008 ED-Media World Conference on Multimedia, Hypermedia & Telecommunications*. Vienna, Austria. June 30 – July 4.
- Zull, J. (2002). *The art of changing the brain*. Sterling, VA: Stylus Publishing.

Biographies



Keith Whittington is an Associate Professor at RIT who believes it is critical for a teacher to actively engage students in the learning process in order to maximize student learning. Several years ago he transformed the way computer programming courses were taught by incorporating active learning techniques into the classroom. His teaching methods significantly increased student retention, grades, and satisfaction in these courses and received an NSF grant to further support his work. He also was the recipient of RIT's 2007 Eisenhart Award for Outstanding Teaching. He has over 40 publications, presentations, workshops, and speaker engagements (including keynote addresses) at various universities, technical conferences, and teaching & learning conferences. He was the co-chair of the 2007 CCSCNE (Computer Science) conference is the current co-chair of the 2008 Teaching Professor conference.



Michael Yacci is a Professor in Learning and Knowledge Management Systems at Rochester Institute of Technology. Dr. Yacci earned his PhD in Instructional Design, Development, and Evaluation from Syracuse University and his current research interests are investigating the limits of mediated cognition and the design of pedagogical agents. Yacci has extensive research publications and industrial and government consulting experience with clients such as Kodak, Xerox, IBM, and the Association for Excellence in Government. He was the recipient of RIT's 2000 Eisenhart Award for Outstanding Teaching.