https://doi.org/10.48009/2_iis_2011_129-134

Issues in Information Systems

Volume XII, No. 2, pp 129-134, 2011

STUDIO-BASED LEARNING IN AN INTRODUCTORY PROGRAMMING CLASS: A COLLABORATIVE STUDY

Sharon Vest, University of Mobile, svest@umobile.edu
Debra Chapman, University of South Alabama, dchapman@usouthal.edu
Leo Denton, University of Mobile, ldenton@umobile.edu

ABSTRACT

Studio-Based Learning (SBL) is an approach to teaching and learning that emphasizes the collaborative application of course skills, critical analysis, and communication. Design-oriented components of SBL include: (1) Construction of a solution, (2) Presentation of the solution, (3) Peer review of the solution, and (4) Modification of solution integrating peer review. This paper describes an implementation and the results to date of the SBL approach at two sister institutions in the introductory programming classes. This experiment is part of a NSF funded CPATH study on the use of Studio-Based Learning in computing classes being conducted among multiple institutions.

Keywords: Studio-Based Learning, Project Based Learning, Peer Review, Critical Thinking Skills, Problem-Solving Skills.

BACKGROUND

Studio-Based Learning (SBL), adapted from architectural education by Hundhausen, *et al.* [6], is an approach applied to the teaching and learning computing that emphasizes the iteration of the skills of computational thinking, critical analysis, collaboration, and communication. Students solve complex design problems collaboratively and present their solutions to peers and instructor(s) for review followed by revision(s) reflective of the review process.

The concept of Studio-Based Learning has been used under different names since the 1800s in the U.S. to refer to a collaborative, mentoring, hands-on approach to teaching and learning. The strategy, in early implementations, focused on master-apprentice relationships in skill training and the arts. Other names, for similar approaches throughout the educational history of the U.S. include the Quincy Systems and the platoon system. The platoon system recognized the role that community played in motivating achievement, a defining characteristic in the Studio-Based Learning strategy described in this writing [9].

Studio-Based Learning evolved from previous studies of the socio-cultural constructivism thread of constructivist learning theory. Principle elements from these previous works included learning by doing, collaborating with the environment (other students, instructors, and external stakeholders), and re-doing until an agreement was reached among stakeholders. As applied to programming projects, this means that the design, output, structure, source code, and other goals were met or exceeded.

Significant enhancements to learning have been found in various studies. According to Boyer & Mitgang, Studio-Based Learning "is really about fostering the learning habits needed for the discovery, integration, application, and sharing of knowledge over a lifetime" [2]. Moreover, Cuff noted the positive effects of peer reviews, called "desk critiques," where the problem solving strategies of the better students influenced the other students' evolving habits throughout the academic period. The behavior exhibited by the better students, observed and noted by their peers, included more iterations and alternative considerations prior to making final solution decisions [3]. Observational data in research conducted by Hundhausen and Douglas of students working collaboratively to visually represent complex problems noted that discussions were more holistic in nature, as opposed to implementation and detail focused; and students were "engaged in more salient discussions about the behavior of their algorithms" [7]. In another study in architectural education, Allen noted that students learned technical skills more effectively when engaged in a problem solving challenge that required the selection and application of those skills and knowledge in the process as opposed to alternative instructional delivery techniques such as lecture or reading assignments with no practical application [1].

SBL, as defined by Hundhausen, *et al.*, [6] can be applied to multiple CS courses throughout the curriculum because (a) it is scalable to all typical computing class sizes, (b) it is adaptable to any course with problem solving and critical thinking as objectives regardless of content or individual teaching style, and (c) it is technology independent.

As Hundhausen, *et al.*, summarized in their NSF CPATH Project Description [8], Studio-Based Learning is defined by four characteristics:

1. Students are required to construct a solution to a complex problem
2. Students must present solution to peers
3. Students must participate in a critique of the solutions of peers
4. Students must respond to criticism by revision or rebuttal.

Of the four, two are key. First, students are given a series of challenging computational design problems to solve individually or collaboratively. Second, students participate in a series of "design crits." These design critiques are review sessions in which students present their evolving solutions to these problems for feedback and discussion [8].

Studio-Based Learning is differentiated from project based learning in several significant ways. First, group responsibilities are required. One of the fundamental objectives is to build a sense of community resulting in a support group that improves all students' levels of knowledge and performance. Loyalty to the group is a powerful motivator for students to persevere to project completion. Second, students are required to present solutions to peers for review. In project based learning, the audience is an optional component. Third, peer evaluation and revision using the process improvement loop is a required component of SBL. Groups must understand and apply critical feedback as demonstrated by improved product after revision.

Project selection for SBL is a critical component for success. The students have no input into the project selection as it is specifically designed to (1) teach fundamental CS concepts necessary to proceed to subsequent more complex concepts; and (2) be of sufficient complexity to challenge a group of students to exercise problem-solving skills in the process. Problem selection is critical, and, if done right, will initially result in short-lived gnashing of teeth among students. If the problem is too complex, students are unlikely to solve the problem adequately or to experience the sense of accomplishment that comes from successfully applying problem solving and critical thinking skills. If the problem is too simple, then problem solving and critical thinking skills are not called into play to successfully complete. "Just right" problems require students to practice and improve problem solving and critical thinking skills. Students are allowed to decide, through collaboration, the "role" they will play in the ultimate solution.

The research questions investigated by all participating faculty in the NSF CPath project [10] are:

1. *Are students learning better from the SBL approach?*
2. *Are students getting more motivated by the SBL approach?*
3. *How does the SBL approach align with ABET CAC accreditation criteria?*

In addition, we wanted to answer one more question through our collaborative efforts related to class size and the effectiveness of the strategy.

4. *Does class size impact the effectiveness of the SBL instructional strategy?*

COLLABORATIVE RATIONALE

After completing a practice run of the strategy while teaching at Honolulu Community College with class of 21 CS and Engineering students, a co-author returned to the mainland still engaged in the NSF project and ready to begin an official research study at University of Mobile (UM). UM is a small private institution with liberal arts education as its core competency with strong recruitment in the areas of Business, Christian Studies, and Music. The number of CIS majors is very small. Currently, the total is 28.

It was felt that the small number of student data points at UM could be problematic in analyzing student outcomes and that a collaborative effort with a sister institution could yield interesting comparative data. Therefore, a pitch was made to the School of CIS of the University of South Alabama (USA), and USA accepted.

The University of South Alabama is a medium size state university with 20,000 students and a strong CIS enrollment. USA has 4 CIS specializations: Computer Science, Computer Information Systems, Information Technology, and Computer Engineering.

STUDY PLAN

Each university chose courses from its introductory programming sequence for the study. Two courses at UM are being used for the study: CIS 265 (Programming I) and CIS 365 (Programming II). One course at USA was used: ITE 285 (Intermediate Programming). Each university would use an initial set of students for a control group and then an experimental SBL group.

In Fall 2010, USA offered ITE 285 in multiple sections using a standard teaching methodology (the control group), and then, in Spring 2011, USA offered the SBL or experimental methodology for ITE 285 in multiple sections.

At UM, CIS 265 in Fall 2010 and CIS 365 in Spring 2011 used a standard teaching methodology (the control group). The SBL approach will be implemented at UM in the same two courses in Fall 2011 and Spring 2012.

Training in the SBL strategy was provided over the summer of 2010. Regular monitoring to insure both programs were delivering the instruction according to the guidelines was conducted. During each semester, multiple collaborative sessions were held to discuss progress, answer questions, and plan for the next phase.

CONTROL GROUP IMPLEMENTATION

The control groups were taught using the usual instructional delivery strategies found most successful in the past. This traditional strategy consists of weekly lectures, small hands-on lab exercises to reinforce key concepts with 4-6 significant programming assignments/projects to demonstrate application of multiple concepts learned. The class met twice weekly for one and a half hours with students earning 3 credits.

SBL GROUP IMPLEMENTATION

To meet the CPATH's SBL guidelines, 50% or more of the class "projects" must be studio based, meaning group assignment, student presentation, and peer review followed by revision. Our plan was to assign three major projects, all of which would be studio-based, while concurrently assigning small, individual, programming assignments as the class progressed through the course objectives. 100% of the three course projects would be studio-based.

DATA COLLECTION AND RESULTS TO DATE

The items for data collection include pre and post tests of course content, pre and post attitudinal and sense of community surveys using the Motivated Strategies for Learning Questionnaire (MSLQ-National Center for Research to Improve Postsecondary Teaching and Learning) and Classroom Community Scale, an end of term questionnaire, course grades and an instructor experience report. There are 51 questions on the MSLQ: Questions 1 through 14 measure motivation and attitude toward the class (Scale 1- 7); Questions 15 through 22 measure learning strategies and study skills (Scale 1-7); Questions 23 through 31 measure class performance (Scale 1-5); Questions 32 to 51 measure the sense of community in the class (Scale 1-5). The range of the 7 point Likert-type scale of the first 22 questions was 1 meaning "not at all true of me" to 7 meaning "very true of me." The 5 point Likert-type scale used for the remaining questions indicated "strongly agree" with a 1 and "strongly disagree" with a 5; therefore, lower scores on questions 23- 51 were indicative of higher scale values. As of this writing, no data is available for UM's experimental classes as these classes will occur in Fall 2011 and Spring 2012, while partial data analysis is available on USA's control and experimental classes.

ITE 285 Fall 2010 and Spring 2011 Data to Date

Two sections of the control ITE 285 class were taught Fall 2010 semester. The total enrollment in the control group at the beginning of the semester was 36. Seven students (19%) dropped the course during the semester. The average score on the

pretests was 23.59 with a posttest average of 71.44. Two sections of the experimental ITE 285 class were taught Spring 2011 semester. The total enrollment in the experimental group at the beginning of the semester was 61. Eighteen students (30%) dropped the course. The average grade on the pretests was 21.76 with a posttest average of 66.32. Experimental group did not show any improvement in posttest scores over the control group; both drop rate and posttest were negative in comparison. The mean scores on 5 of the 7 weighted grade items, Assignments, Quizzes, Project 1, Project 2, Exam 1, Exam 2, Exam 3, of the experimental group were higher than the control group, but only two were statistically significant, Quizzes and Project 1. The final overall average for the control group was 70 and 72 for the experimental group. This difference was not statistically significant.

Results from the end of term questionnaire may provide insight not reflected in other data analyzed to date. The questionnaire posed questions related to the students' perception of their experiences in the course. Both groups were given a survey with 11 questions in common plus additional questions specific to whether the group was control or experimental. Both groups answered similarly to Question 1, *Did you enjoy this course more or less than what you expected at the beginning of the course?*, 22% of the control group and 26% of the experimental group responded that they enjoyed the course more with 33% and 31%, respectively, responded less. To Question 5, *Having taken this course, are you looking forward to taking more courses in computing?*, both groups responded with 83% affirmative and 17% negative. Four of the common questions were answered similarly for both groups. Several of the other questions suggest differences that hold promise for more research. To Question 2, *Did you learn more or less than what you expected at the beginning of the course?*, 33% of the control group responded to each of the three options, more, less and neutral; 66% of the experimental group responded with more, 6% less and 10% neutral. To Question 10, *Has it been easy in this course to learn about how other students solved programming projects/assignments/labs?*, 44% of the control group and 63% of the experimental group responded yes with 56% and 37%, respectively responding no. To question 11, *Did you review and provide written feedback on other students' projects/assignments/labs?*, 22% of the control group and 86% of the experimental group responded affirmatively. To Question 12, *Did you find the process of reviewing and providing written feedback on others' work helped your learning in this course?*, 0% of the control group and 54% of the experimental group responded affirmatively, with 22% and 31%, respectively, responding negatively.

Of the additional questions posed to the experimental group, the majority of students answered affirmatively to all, including the following:

1. *Did you receive helpful feedback during your group activity?* (74%)
2. *Did you find being an audience member when other students presented their work helpful to your learning in this course?* (74%)
3. *Did you ask questions and/or provide feedback during other students' presentations?* (57%)
4. *Did you find asking questions and/or providing feedback helped your learning in this course*? (74%)
5. *Did you like working in groups?* (66%)

Of the 2 additional questions posed to the control group, the responses include:

1. *Did you work in groups?* (no--100%)
2. *Do you wish you had worked in groups?* (66.7%--yes; 33.3%--no)

No attitudinal or sense of community survey data is available for the ITE 285 control or experimental groups at this time. The research committee is currently developing the code book for the responses and that data should be available in 2012.

CIS 265 Fall 2011 Data

The first control group at UM, CIS 265, Programming I, started with 7 students enrolled Fall 2010. No students withdrew during the semester, but two students failed the course. All seven scored 0 on the pretest. The posttest average was 52.85 of a possible 100 points, with a high of 90 and a low 0. Both students who failed the course made zeros on the pretest; therefore, the average posttest score of the students who received course credit was 74. The final course average for all students was 65.57, and 87.8 among those who received credit.

The second control group at UM, CIS 365, Programming II, started with 7 students enrolled Spring 2011. One student withdrew during the semester, and another student failed the course. All students made a zero on the pretest. The posttest average was 59.65 of a possible 100 points, with a high of 84 and a low 16. The average posttest score of the students who received course credit was 68. The final course average for all students was 76.30, and 82.5 among those who received credit.

Survey data on the CIS 265 or 365 control groups will be available in 2012.

OTHER PROJECT INSTRUMENTS

SBL Project management and evaluation instruments were developed for use in the experimental courses and discussed during planning sessions between the two institutions and are included in the appendix. The instruments included:

1. Communication Protocol for Team Projects.
2. Project Check List
3. Project Peer Inter-Team Evaluation
4. Project Peer Intra-Team Evaluation
5. Final Project Report Requirement

The Communication Protocol is a proactive tool used to guide students into planning how they will collaborate early in the project timeframe. The Project Checklist provides grading criteria to students with assignment details to communicate learning objectives and deliverable expectations. The Inter-Team and Intra-Team Evaluations are used during peer reviews. The Final Project Report formalizes the closure of the feedback loop in the learning process.

CHALLENGES AND RECOMMENDATIONS

The most difficult parts of the SBL implementation are problem design and group membership selection. Another important criterion for success is the development of clear evaluation techniques and instruments that are communicated effectively to the students prior to the first submission. Research exists that provides recommendations for grouping students for maximum synergetic effects. Some studies advocate assigning members of each group based on similar levels of knowledge and skill [4].

As with all instructional strategies, structure and clear evaluative instruments are crucial. Students who clearly understand expectations usually work toward achieving them. Peer review instruments and weighted grading criteria were developed and discussed prior to the first project assignment with repeated review of expectations until the final presentation date. As learned from previous course experiences, pair programming techniques in the weekly labs work well and provided an excellent complement to the SBL approach.

In reviewing the experience, selection of the group projects caused the most anxiety and is a key to achieving positive results from the SBL process. The problem must be sufficiently complex to require a team of students to adequately solve it but not too complex. Individual students, in the process, should be forced to engage collaboratively in critical thinking and problem solving. Group assignment is an area where current research findings can be applied for improvement.

FUTURE PLANS

Data from both the control and experimental ITE 285 classes will be reviewed and results reported by August 2012. Data from the CIS 265 and CIS 365 experimental groups will be collected over the 2011-2012 academic terms and will be available for comparison by August 2013. Faculty will continue to improve skills in the SBL instructional delivery strategy for effective comparison to the already much rehearsed traditional strategies.

REFERENCES

1. Allen, E. (1997). Second Studio: A Model for Technical Teaching. *Journal of Architectural Education* , 39-43.
2. Boyer, E. L. and Lee Mitgang (1996). *Building Community: A New Future for Architecture Education and Practice.* Princeton: The Carnegie Foundation for the Advancement of Teaching.
3. Cuff, D. (1992). *Architecture: The Story of Practice.* Cambridge: MIT Press.
4. Gunderson, D. E.and Jennifer Moore. (2008). Group Learning Pedagogy and Group Selection. *International Journal of Construction Education and Research* , 34-45.
5. Hendrix, D., Myneni, L. Narayanan, H., and Ross, M. (2010). Implementing Studio-Based Learning in CS2. In *Proceedings of the 42st ACM Technical Symposium on Computer Science.* Education (SIGCSE 2010), Milwaukee:ACM.
6. Hundhausen, C., Anukrati Agrawal, Dana Fairbrother, and Michael Trevisan. Does Studio Based Instruction Work in CS 1? An Empirical Comparison with a Traditional Approach. *Proceedings of SIGCSE'10,* Milwaukee:ACM.
7. Hundhausen, C. and Sarah Douglas. (2000). Using Visualizations to Learn Algorithms: Should Students construct Their Own, or View an Expert's? *Proceedings of the 2000 (IEEE) International Symposium on Visual Languages* (pp. 3-8). Piscataway: IEEE.
8. Hundhausen, C., Hari Narayanan and Martha Crosby. (2007-2012). NSF CPATH II Project Proposal, Collaborative Research: CPATH II: Broadening Studio-Based Learning in Computing Education.
9. Lackey, J. (1999, August 2). *A History of the Studio-Based Learning Model.* Retrieved October 5, 2011, from http://www.edi.msstate.edu/studio.htm
10. NSF CPATH II. (2007, March 1). Studio-Based Learning. Retrieved March 3, 2008, from Studio-Based Learning Commuinity: http://iis.cse.eng.auburn.edu
11. Wilson, J. and Mosher, D.(Summer 1994). The Prototype of the Virtual Studio Classroom. *Journal of Instruction Delivery Systems*.