JET The Journal of Effective Teaching
an online journal devoted to teaching excellence

Promoting Technology-assisted Active Learning in Computer Science Education

Jinzhu Gao[1] and Jace Hargis
University of the Pacific, Stockton, CA 95211

Abstract

This paper describes specific active learning strategies for teaching computer science, integrating both instructional technologies and non-technology-based strategies shown to be effective in the literature. The theoretical learning components addressed include an intentional method to help students build metacognitive abilities, as well as improve on their self-efficacy, both inside their chosen discipline and connections to other interdisciplinary topics. The results indicate that students are very open and eager to embrace novel ways to become engaged in learning in the area of computer science.

Keywords: Active Learning, Flip Camcorder, Emerging Technologies, Metacognition, Self-Efficacy.

1. Introduction

There are many different methods to teach computer science courses. Traditionally teacher-centered didactics are widely used in the United States and many parts of the world. In recent years, some learner-centered approaches that encourage active learning have been explored and implemented. Such pedagogical innovation is so important that the University of the Pacific (Pacific) states its mission as *"to provide a superior, student-centered learning experience integrating liberal arts and professional education and preparing individuals for lasting achievement and responsible leadership in their careers and communities."*

Computer Science (CS) is a discipline that explores how to use computers to solve real world problems. Therefore, it is crucial to create an active learning environment to improve students' comprehension and retention of material, allow students to take control and regulate their own learning, and eventually empower them with necessary skills to solve problems outside of the classroom. Studies also show that "active learning is especially effective for CS students who tend to be visual/intuitive learners (Briggs, 2005)." To achieve our goal of keeping students engaged in an active learning environment, throughout the teaching and learning process, it is important to continue our own inquiry, as CS instructors on how people process information and subsequently learn in a deep and authentic way: How can we effectively use appropriate, functional emerging instructional technologies to address this goal of applied understanding? Will spending consid-

[1] Corresponding author's email: jgao@pacific.edu

erable time and energy attending to an active learning environment address this question? What is the extent of freedom and autonomy which students can be provided to encourage an active and empowering learning experience?

In the Introduction, Section 1, we have provided a brief background on this study. In Section 2, we review previous efforts and literature in developing pedagogical approaches that promote active learning. The strategies which we were used are described in Sections 3, the findings are shared in Section 4, our conclusions are summarized in Section 5, and the future work is described in Section 6.

2. Literature Review

The use of active learning techniques are increasing in many disciplines of higher education and the field of computer science and engineering is no exception. In its basic form, active learning is defined by many and in many ways; however, most agree that there are several essential components. First, active learning is considered as anything other than passively listening to lecture, where students apply material to “real life” situations (Paulson & Faust, 2002). Second, in a typical active learning environment, “students must talk about what they are learning, write about it, relate it to past experiences, and apply it to their daily lives. They must make what they learn part of themselves (Chickering & Gamson, 1987).” Third, active learning should be supported because “what students learn is greatly influenced by how they learn, and many students learn best through active, collaborative, small group work inside and outside the classroom (American Association for the Advancement of Science, 1989, 1990; National Research Council, 1995, 1996; National Science Foundation, 1996; Springer, Stanne, Donovan, 1998).” The connection between collaborative/cooperative learning and relevance in applying these skills towards an authentic, contextual scenario becomes a key, especially in areas such as engineering and computer science. Research shows that “an active learning environment can help by sharing experiences, and providing guidance and support (Johnson, Johnson, & Smith, 1991).” Fourth, as Briggs (2005) points out, in a rapidly changing field as Computer Science, “students tend to be active and visual learners” and it is beneficial to provide a learning environment where students can interactive with the material. Over the years, various strategies have been developed by CS instructors to promote active learning (McConnell, 1996; Chen, Christel, Hauptmann, & Wactlar, 2005; Anderson, Anderson, Davis, Linnell, Prince, & Razmov, 2007; Schweitzer & Brown, 2007). Active learning can incorporate many instructional methods, such as collaborative/cooperative, project/problem-based learning, role play, debates, etc. and even the use of functional emerging instructional technology teaching tools. Lindquist et al. (2007) demonstrates how mobile phones can be used to “broaden and enhance the use of active learning in large classrooms.” Dumova (2008) studies the use of digital video assignments as an active learning tool. Another relatively new technology teaching tool is the Flip camcorder device (http://www.theflip.com/), shown in Figure 1, which is an easy-to-use, pocket-sized HD camcorder with simple user interface, built-in memory, and a flip-out USB arm that plugs directly into the PC or Mac. Using the Flip camcorder, instructors are able to quickly gather real-time evidence of students’ conceptual learning styles specific to their discipline. In one recent study using the Flip camcorder, the major finding was the "breadth of generalizability for the use of this tool across disciplines for increasing student engagement (Hargis & Marotta, 2010)."

Figure 1. The Flip camcorder device

3. Methods

Throughout our teaching practice, we have used several innovative pedagogical approaches to achieve our goal of promoting technology-assisted active learning in CS education. The guiding principle is to engage students in the process of active learning so that they can successfully meet learning goals.

3.1 Using Advanced Teaching Tools

Various online learning management systems, such as Blackboard, Sakai, Moodle, myCodeMate, and WileyPLUS, have been developed for delivering, tracking and managing training and education. These systems help instructors create a virtual classroom that allows students to learn anytime, anywhere after class, which greatly enhances the communication between instructors and students. Easy access to a wide variety of digital learning content makes both teaching and learning more efficient and effective. Some systems provide interactive exercises that help students learn how to apply techniques to solve real-world problems. For example, we use myCodeMate, the completely web-based, textbook-specific homework and programming resource for the introduction to programming course, to show students in step-by-step manner the way to write and test programs. It also automatically gives quick feedback on the students' performance, from which instructors are able to evaluate learning outcomes and immediately adjust teaching pace and strategies to achieve the best teaching result.

Since the spring of 2009, our classes began using portable Flip video camcorders in many of our activities. We found that, due to its easy-to-use interface and the flip-out USB arm, Flip video camcorders are very convenient for both instructors and students, in and outside of class- rooms. With limited efforts and time, Flip video camcorders can help us easily capture and share low-resolution video via e-mail and the Web, which greatly enhance the active learning experience. For example, in the *Application Programming* class (23 students), students in each group were required to work on a course project that emphasizes team efforts and member contributions. The success of a student project largely depended on group activities including group discussions and weekly presentations. Using Flip video camcorders, we were able to record these activities, which documented the whole process of software development for each team and played an important role in project management. Throughout the semester, students could review these activities whenever necessary and instructors could use the video clips to evaluate the progress of all course projects. As an extra bonus, we found out that these video clips could be especially useful for students who had to miss the discussion or students who need more time to think to catch up. To monitor project progress, each team needed to give weekly mini-presentations. We recorded these mini-presentations and played them back immediately to all students. When students knew their talks would be videotaped, they became more serious and tried to look more professional. Some students who were usually quiet in class actually spoke out in front of the Flip camcorder, which was really a surprise. We believe these particular types of activities might be helpful to students who might posses a marginal social anxiety disorder or students who are less confident in public speaking. When we used Flip video camcorders in the classes such as *Introduction to Computer Science* (10-20 students) and *Design and Analysis of Algorithms* (20-30 students), we noticed that, by viewing different solutions students used to solve a single problem recorded by the camcorders, students could easily review solutions given by other students and gain better understanding of data structures and fundamental concepts, which in turn improved their programming skills.

Every year, we invite our CS alumni to visit our class for networking between students and alumni as well as gathering input from alumni on professionalism. By watching the video clips we prepared, our alumni have opportunity to observe some of our class activities and help us evaluate our efforts on improving teaching effectiveness. Their suggestions are also recorded with the portable Flip Video Camcorder so that we could share their insights with our future students as well as create a clearinghouse of archived video information over the years. We believe these video clips will have positive impact on our students.

3.2 Introducing State-of-art Technologies and Facilities

Since computer science is a dynamically changing field, students should be exposed to state-of-art technologies and facilities throughout their learning experience. From our experience, the more students know, the more confident they become and the more active they are in future learning. It is especially important to senior students as they prepare for their job interviews. To help students be prepared for a quickly changing world, we develop various class activities and course projects that get students familiar with advances in state-of-art technologies. For example, when teaching courses such as Operating Systems in a traditional way, instructors tend to focus more on lectures and lab assignments. Students usually feel bored during long lectures and sometimes become less motivated to finish lab assignments. However, to achieve teaching goals, instructors still have to make sure students understand course materials. As we know CS students tend to be active, we asked students to form groups and each group was required to lead regular class discussions on one operating system they felt most interested. We instructors can then introduce terms and techniques in between their discussions. To document their discussion and presentations, Wiki pages and recorded video clips were used, which was proven to be very effective in engaging students. We also found that it can actually be done without much effort as students themselves use different operating systems and they are all very open-minded to new technologies. Students felt more involved in the classroom. They were also willing to spend a lot of time after class to collect information for these operating systems and learnt many things that are usually not covered in textbooks. At the end of the semester, they were able to evaluate their systems from both user's and developer's point of view, which was really a big plus.

To help students gain more practical experiences and prepare them for real-world challenges, when teaching system courses, we give students opportunities to install, use, and configure various systems including virtual machines and supercomputers. By comparing the running performance on different systems, students can easily see both strengths and limitations of these systems. One student, who took the parallel computing class previously, told the instructor (Gao) that, during a job interview, the interviewer was impressed by the fact that the undergraduate student was able to write parallel programs on supercomputers. And some students were not successful in their interviews because they did not have similar training experiences.

3.3 Promoting Students' Creativity

"You cannot teach a man anything; you can only help him to find it within himself."
- Galileo Galilei

One of the many challenges in teaching is that students are not uniformly interested in the topics being discussed, and students in any one class generally maintain a wide range of interests. We, as instructors, must be aware of this diversity and learn to engage all the students. Our teaching should be able to cultivate the students' imagination and eventually retain their inherent creativity. To achieve this goal, assisted with various technologies, we also strive to give students more freedom on course projects. We encourage students to be open-minded and creative, especially when they learn new technologies.

For example, in the *Distributed Computing* class (12 students) during the fall 2009 semester, students had freedom deciding which projects they would like to work on and could even develop their own projects. We had 7 groups of 1 to 4 students, each working on a distinct project. After the semester was over, three groups of students enjoyed their projects so much that they decided to continue their projects as independent study or senior project during the spring 2010 semester. With limited budget, one group of students even made great effort to build an up-to-date LCD panel based display system for our CS department. In this active learning process, we could see most of our students were able to move "beyond the textbooks" and became more creative and confident.

4. Results

To demonstrate the effectiveness of active learning pedagogical approaches, in our previous teaching experiences, we deployed both the traditional teaching methodology and the active learning methodology. Before presenting results, we first describe differences between these methodologies below.

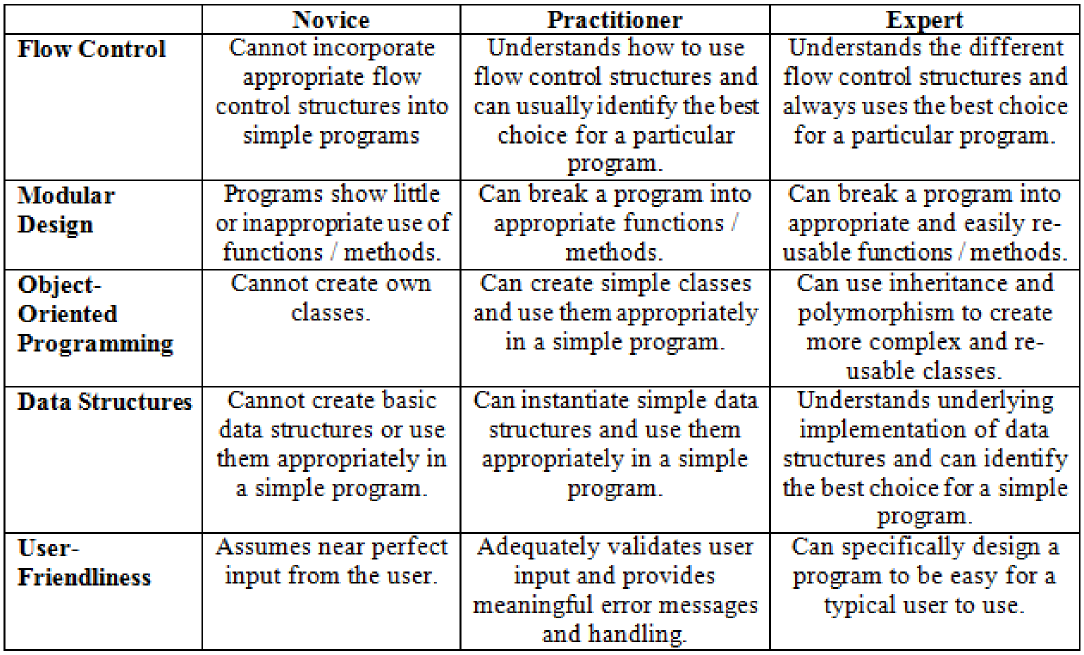
- *Traditional Teaching Methodology*: Teachers transfer knowledge to students mainly through formal lectures and reading assignments. Such approach has a long history and has been widely used. However, class activities are typically done in a reflexive, mindless manner, which limits students' higher level thinking. As students' attention tends to drop off automatically after 10-20 minutes, students frequently think about things unrelated to the lecture content for most of the time during lectures. In addition, teachers' teaching pace is difficult to control in order to satisfy each student's need.
- *Active Learning with Less Teacher-Guidance*: It uses the constructivist learning model in which students construct rather than receive knowledge. The key in this model is the connections between conceptual frameworks which are developed or constructed from prior knowledge. Teachers provide learning goals and give project assignments. After that, teachers observe how students conduct activities in groups and, when necessary, answer questions. This approach puts students in the center of learning and students' curiosity drives learning, which encourages students to build skills through experience.
- *Active Learning with More Teacher-Guidance*: It is also based on the constructivist learning model. However, teachers get more involved in activities that engage students in active learning. Such a student-centered approach allows teachers to identify each student's capability and individuality and help students to develop skills based on their past experiences and learning. When we take into account each student's learning needs and preferences, the student tends to play an active role in learning and easily develop an interest for the subject. The formation of such enduring attitudes can be very important for future study.

When teaching *Introduction to Computer Science*, the instructor (Gao) used the traditional methodology in one semester and the active learning methodology using advanced teaching tools in the following semester. As it is the introductory computer science course and half of the students are not computer science major, the instructor (Gao) chose to provide more teacher guidance. The outcomes are shown in Table 1. The instructor (Gao) collected the results from both teachers' and students' point-of-view. On a 5-grade

Table 1. This table compares the teaching and learning outcomes of the same course, "*Introduction to Computer Science*" (10-20 students), using two pedagogical approaches: teacher-centered and student-centered.

Evaluator	Evaluation Item	Teacher-Centered	Active Learning
Teacher	Students meeting learning requirements	70%	92%
Teacher	Average attendance	60%	95%
Student	Average evaluation on the instructor	4.13 out of 5	4.6 out of 5
Student	Average evaluation on the course	3.91 out of 5	4.31 out of 5
Student	Average evaluation on the labs	3.25 out of 5	4.47 out of 5

Figure 2. Rubrics for *Introduction to Computer Science* at Pacific.



	Novice	Practitioner	Expert
Flow Control	Cannot incorporate appropriate flow control structures into simple programs	Understands how to use flow control structures and can usually identify the best choice for a particular program.	Understands the different flow control structures and always uses the best choice for a particular program.
Modular Design	Programs show little or inappropriate use of functions / methods.	Can break a program into appropriate functions / methods.	Can break a program into appropriate and easily re-usable functions / methods.
Object-Oriented Programming	Cannot create own classes.	Can create simple classes and use them appropriately in a simple program.	Can use inheritance and polymorphism to create more complex and re-usable classes.
Data Structures	Cannot create basic data structures or use them appropriately in a simple program.	Can instantiate simple data structures and use them appropriately in a simple program.	Understands underlying implementation of data structures and can identify the best choice for a simple program.
User-Friendliness	Assumes near perfect input from the user.	Adequately validates user input and provides meaningful error messages and handling.	Can specifically design a program to be easy for a typical user to use.

scale, students evaluated the instructor (Gao) from 15 aspects including “Ability to keep students interested/motivated.” The instructor (Gao) relied on the rubrics shown in Figure 2 to evaluate whether a student meets learning requirements (or at least at the Practitioner level) or not. By analyzing the results, we realize that the approach promoting active learning is more effective in keep students engaged as it allowed instructors to build a productive working relationship with students and students feel more comfortable and active throughout the learning process. From our experience, for introductory courses, more teacher involvement is desirable. For higher level courses, instructors could give students more freedom, especially when working on creative projects.

4.1 Qualitative Testimonials

There were many comments on the method of instruction from a wide variety of students. These included:

- “I learned a lot about programming and I am confident in my coding now.” - Student from
- *Introduction to Computer Science.*
- “The interviewer was so impressed that I knew parallel programming.” - Student from *Distributed Computing*.
- “I really learned a lot of useful algorithms. In fact I talked about some of the sorting algorithms we learned in class during my co-op interview with NVIDIA, and I think that actually helped me get the job” - Student from *Design and Analysis of Algorithms*.

- "The instructor teaches students a new way of thinking. She is willing to help students without giving the answers away easily." - Anonymous student from *Data Structure*.
- "The instructor creates a good learning environment and makes class fun. She provides useful labs and small projects that are able to promote valuable learning experiences. For example, the file system lab was very helpful in demonstrating how an effective file system can benefit our work. Involving class in questions at the end is also good, which keeps the class engaged." - Anonymous students from *Advanced Operating Systems*.

4.2 Other Results

In addition to the qualitative results, we consciously and unconsciously were able to gather critical pedagogical information for our planning and deployment of instruction. These include improved lesson planning, formative assessment, integration of technology as well as other disciplines in the course, gathering student feedback and subsequently providing real-time remediation opportunities, as well as other intangible outcomes which influenced both the instructors and student affective domain, namely in the area of dispositions, emotions and philosophy towards teaching and learning.

Specifically, data observed which addresses each of these phenomena include:

- Lesson plans were prepared in an intentional manner prior to the class, integrating active learning theory and included the Topic, Time required for each activity, Student Learning Outcomes (what we wish the learner to know or be able to do and under what conditions and are measurable), Standards, Activities and associated forms, Materials, Academic Content, Procedures (specific details with key higher level inquiry-based questions and experiences, which include an opening hook, middle, and close), Follow-up and Assessment, both formative and summative, as well as cited resources.

At the end of each semester, we ask students for their opinion on the textbook, the projects, and the teaching pace. We also ask for valuable suggestions. This information has been very useful for us to make adjustments to teaching plans, especially when we need to teach the same course again in the near future. When we prepare lessons for the first time, we collect similar information from previous instructors and students.

- Formative assessment: To check students' learning outcomes, we observe students as they ask or answer questions. In programming courses, we observe the way students write pro- grams to solve problems. The student reaction from various class activities is also used so that instructors can adapt the pedagogical approaches to meet students' needs. From the observations, we used a rubric, such as the one shown in Figure 2, which contained specific criteria as to the type of behaviors and outcomes, which we believed critical in the process.

- *Integration of technology*: Frequently, we utilized instructional technology to improve the communication between the instructor and the students. For example, we created a Forum area and a Wiki on the Sakai Collaborative Learning Environment to allow both the instructor and the students to share their opinions on certain topics. We also used Portable Flip video camcorders to record the classroom activities and both instantly previewed for real-time reflection, as well as posted on-line for asynchronous viewing and remediation. The information can help the instructor make necessary adjustments.

- *Real-time remediation opportunities*: Throughout the class period, we asked students to solve some problems to prove they understand basic concepts or algorithms. We then checked on each student's solution and, whenever possible, give hints and encourage them to make corrections or improvements. We used this approach in most of our computer science courses and find that it speeds up students' understand process and greatly improves students' problem solving capability.

- *Affective domain*: The affective domain typically includes factors such as student motivation, attitudes, perceptions and values. In order to enhance active learning, the instructor should consider the affective domain in planning courses, delivering lectures, developing class activities, and assessing student learning. We achieve this by talking to students who took the same course in previous years and, before or after each class, frequently checking on students' feeling through relaxed conversion. For example, most of students felt very frustrated with one Operating System lab mainly because they did not know how to implement the menu. After having conversion with students, considering that the menu implementation is actually not the focus of the lab, the instructor (Gao) decided that the menu is not a required item and, students who still implemented the menu could get extra credits. Such adjustment immediately removed students' fear and allowed students to focus on the major points of the lab. As a result, all students were able to finish the lab with great enthusiasm.

5. Conclusion and Future Work

In this paper, we described several innovative pedagogical approaches we used in our CS classes to promote technology-assisted active learning. Advanced teaching tools are used to assist both teaching and learning. Class activities are developed in a way that promotes students' creativity. State-of-art technologies and facilities are introduced to broaden students' horizons and career opportunities. From our experience, we find that, for most computer science courses, these innovative approaches should be used together to achieve the optimal result.

5.1 Connection to Learning Theories

Active engagement in the classroom is not a new event in higher education, even in the discipline of computer sciences (McConnell, 1996; Chen, Christel, Hauptmann, & Wactlar, 2005; Anderson et al., 2007; Schweitzer & Brown, 2007). Frequently, we have asked students to participate in hands-on, minds-on activities during and after class developing programs, simulations and more. The intentionality of this research was to introduce methods to help students become more engaged in their own internal processing of information, including metacognition and self-efficacy. Providing a quick-capture video system (Flip camcorders) encourage interactivity, discussions, connections and reflection of each students thoughts, behaviors and visuals of their performance. As the example in the Operating System class in Section 4.2 demonstrates, overt conversations on what students were thinking, how they were processing and ultimately how this modified their thoughts were noted. These events occurred especially in the venue of learning, as they captured the major conceptual frameworks which frequently changed the way they studied and completed their assignments.

5.2 What We Learned

One of the major walk-away messages for instruction was the possibility that we can promote active learning and improve students' interests in learning through innovative pedagogical approaches. Although significant literature exist in the arena of active, hands-on/minds-on learning, we now understand more about how our students process information and the directions required to continue active engagement and subsequent assessment of their learning. Developing more formative assessment, avenues of continual feedback and a mechanism to address the feedback in a manner which both reflects and addresses their needs has become a critical agenda for our instructional style.

Secondarily, we were able to fully discover the potential of using advanced teaching tools and the benefits of adopting student-center approach. Therefore since deployment of the Flip camcorder worked well, we are now investigating other possibly teaching tools to integrate into the classroom. For instance, there are student response systems, Web 2.0 tools, netbooks, various electronic measurement probes and possibly even Wii systems which might enable more advanced student interaction and assessment.

In addition, students were observed to be more involved in an active learning process. Their creativity was also recognized through different class activities, which in turn improve their self-awareness and self-confidence. Historically, we have not attended to or measured creativity in this discipline. Using the technology teaching tools, it was more efficient to deploy and students produced material, which attended to their style and ultimately allowed us to make direct connections between their creativity and the content discovered. This approach empowered students to address their own information processing in unique and individual ways, which at first confused, but ultimately addressed many of their needs, that had been traditionally ignored.

Finally, the instructor (Gao) had opportunity to practice different ways of teaching and learning from the experience. From the student feedback, the instructor (Gao) is able to further improve teaching through improved lesson planning, class activities and integration of other instructional technology. Although there is significant time involved in exploring and learning the use of additional teaching tools in the classroom, the instructor (Gao) discovered that enlisting the assistance of the university Center for Teaching and

Learning as well as her colleagues could result in an efficient method for receiving purposeful information on teaching and learning as well as a scholarly approach to teaching.

5.3 What We Plan to Do Next

The next major steps include repeating this use of Flip Camcorders in other classes, collecting empirical pre/post-assessment data on how students performed on a particular task or assignment, practicing other teaching tools, such as interactive syllabi and classroom preparation sheets, developing our own tools that support active learning and teaching, and conducting study on key causes that affect active learning. In this paper, we limit our discussion to computer science education. However, in many of our classes including *Introduction to Computer Science* and *Application Programming*, about half of our students are actually in majors such as engineering, business, applied math, and physics. Based on our discussion with these students, we expect our strategies would also work well in many other disciplines. To verify this, we will coordinate with instructors of other disciplines and conduct a more comprehensive study that involves multiple disciplines and multiple instructors.

5.4 How We Believe This Will Assist Students' Future Learning

Ultimately, we believe that students will need to access their metacognitive abilities to succeed in future courses in our programs. Programs such as engineering and computer science require a strong sense of self-regulating ability, which students have gained from this experience and will be able to reflect upon, possibly even utilizing some of the same techniques developed in this course to master future concepts.

In addition, there is a capstone requirement for students to graduate and in this sense, the type of learning environment offered in this course, will enable them to sustain self-reflection and create the required connections between the course experiences which they have been offered, the text, internships, and other opportunities. The capstone focuses on active-learning to improve problem solving capability and creativity of computer science students, which can be tied directly to the outcomes of this project by practicing a combination of several innovative pedagogical approaches.

5.5 Down the Road: Connections to Future Careers

Our goals for this project were lofty in that we believe our students will be able to reflect on these experiences along their academic career path and beyond into the workplace. The connections between this project and our departmental goals/missions are the practice of active learning and student-centered pedagogical approach that helps prepare students to contribute to the computing profession and provide students a basis for continued study, graduate student, and growth in their fields. The connections between the department vision and escorting students into a viable, healthy workplace include providing undergraduate education in computer science which features current and emerging technologies and experiential learning. We truly believe that studying computer science is about developing a way of thinking, an approach to solving problems. By promoting active learning, we are able to help students achieve their learning goals and build skills that last a lifetime.

References

American Association for the Advancement of Science (1989). *Science for all Americans: Project 2061*. New York: Oxford University Press.

American Association for the Advancement of Science (1990). *The liberal art of science: Agenda for action*. Washington, DC: American Association for the Advancement of Science.

Anderson, R., Anderson, R., Davis, K., Linnell, N., Prince, C., & Razmov, V. (2007). *Supporting active learning and example based instruction with classroom technology*. ACM SIGCSE Bulletin, 39(1), 69–73.

Briggs, T. (2005). *Techniques for active learning in CS courses.* Journal of Computing Sciences in College, 21(2), 156–165.

Chen, M., Christel, M., Hauptmann, A., & Wactlar, H. (2005). *Putting active learning into multimedia applications: Dynamic definition and refinement of concept classifiers*. Proceedings of the 13th Annual ACM International Conference on Multimedia, (pp. 902–911).

Chickering, A., & Gamson, Z. (1987). *Seven principles for good practice*. American Association Higher Education Bulletin, 39, 3–7.

Dumova, T. (2008). *Using digital video assignments as a tool for active learning*. The International Journal of Learning, 14(12), 63–71.

Hargis, J., & Marotta, S. (2010*). Using FLIP camcorders for active classroom metacognitive reflection in higher education.* Journal of Active Learning in Higher Education, 11(2).

Johnson, D. W., Johnson, R. T., & Smith, K. (1991). *Active learning: Cooperation in the college classroom*. Edina, MN: Interaction Book Company.

Lindquist, D., Denning, T., Kelly, M., Malani, R., Griswold, W., & Simon, B. (2007). *Exploring the potential for mobile phones for active learning in the classroom*. Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education, (pp. 384–388).

McConnell, J. (1996). *Active learning and its use in Computer Science*. ACM SIGCUE Outlook, 24(1-3), 52–54.

National Research Council (1995). *National science education standards*. Washington, DC: National Academy Press.

National Research Council (1996). *From analysis to action: Undergraduate education in science, mathematics, engineering, and technology*. Washington, DC: National Academy Press.

National Science Foundation (1996). *Shaping the future: New expectations for undergraduate education in science, mathematics, engineering, and technology*. Washington, DC: Report by the Advisory Committee to the National Science Foundation Directorate for Education and Human Resources.

Paulson, D., &Faust, J. (2002). *Active learning for the college classroom.* http://chemistry.calstatela.edu/Chem&Bioch/active/main.htm. .

Schweitzer, D., & Brown, W. (2007). *Interactive visualization for the active learning classroom*. ACM SIGCSE Bulletin, 39(1), 208–212.

Springer, L., Stanne, M.E., & Donovan, S. (1998). *Effects of cooperative learning on undergraduates in science, mathematics, engineering, and technology: A meta-analysis.* University of Wisconsin-Madison, National Institute for Science Education, *Review of Educational Research*.