AN EXPERIENCE IN COLLABORATIVE LEARNING: OBSERVATIONS OF A SOFTWARE ENGINEERING COURSE

Heidi J. C. Ellis[1]

Abstract *¾ In this paper, I describe my experiences in teaching an advanced graduate-level software engineering course using a collaborative approach. Software Engineering II (SEII) is the second of two graduate software engineering courses taught at Rensselaer at Hartford, a branch of Rensselaer Polytechnic Institute that provides professional Master's degrees in Computer Science. During the past three offerings of SEII, I have attempted to create a collaborative learning environment which would support both peer-based learning and life-long learning through the employment of such techniques as cooperative construction of mental models, brainstorming, and collaboration by example. This paper provides details on the techniques that were used in SEII, describes my observations on the constraints and successes of my approach, and outlines some thoughts for future directions for SEII.*

Index Terms ¾ Collaborative learning, life-long learning, peer-based learning.

INTRODUCTION

This paper reports on experiences that I gained during the teaching of three semesters of an upper-level graduate course in software engineering. Software Engineering II (SEII), is a topics course typically taken by Master's degree students towards the end of their studies at Rensselaer at Hartford. During the past three years, the course has included an increasingly large component on distributed object technology. As a result of the rapid change occurring in the field of middleware and the amount of effort required to keep abreast of these changes, I desired to leverage students' existing experience with the technology while maximizing the flow of knowledge both among the students and between the students and the instructor. A primary goal of implementing these approaches to teaching SEII was to increase the collaborative learning that occurred both inside the classroom and beyond its boundaries. This paper presents the results of that effort. This paper examines the content and instructional goals of SEII, then describes the techniques used to achieve these goals, and discusses my observations of the success of the various techniques. I begin by providing an overview of Rensselaer at Hartford, its students, and their ambitions.

The Rensselaer at Hartford Student

Rensselaer at Hartford (RH) is a branch of Rensselaer Polytechnic Institute (RPI) located in Hartford Connecticut. RH's primary mission is to provide application-oriented Master's degrees in Computer Science, Engineering, and Management to working professionals. RH has a total student enrollment of around 2000 students, 450 of which are in Computer Science. A typical student at RH is employed full-time and attends classes on a part-time basis. Student ages range from middle twenties into the fifties and students enter the Computer Science program with a spectrum of backgrounds that include undergraduate degrees in computer science, accounting, mathematics, and engineering. Since most of our students work full-time, classes are taught one day a week from 5:30 to 8:30 in the evening.

Students in the Computer Science program at RH have some clearly identifiable goals which impact and shape their learning process. RH students are highly motivated and have a strong desire to learn. Perhaps the most fundamental goal of RH students is to gain knowledge that will broaden their knowledge base for future career moves. Students are interested in learning not only the core knowledge about a subject, but also the most recent advances in the technologies that support that knowledge

Another goal of RH students is the aim to directly translate material learned in the classroom to on-the-job employment of that knowledge. One fundamental reason that students attend RH is that they know that they will receive education that can be immediately and directly applied to their current projects in the workplace.

In addition to the identified goals, students at RH have several other characteristics that impact their learning experience. Since most of our students work full-time, they have only a limited amount of time to accomplish their learning objectives. They desire a learning experience that provides them with the tools and ability to gain new knowledge quickly and efficiently. However, students are also very willing to spend time researching in a new topic area if they are convinced that the information can and will be of direct and immediate use to them.

Another characteristic of RH students is that they are geographically disperse. We attract students from three states, and students have been known to travel up to two hours one way to attend classes. The impact of this distribution is that most students are highly unlikely to interact or even encounter their classmates during the week unless provided with a convenient communication mechanism and a good motivation to use such mechanism.

[1] Dept. of Computer and Information Sciences, Rensselaer at Hartford, 275 Windsor St., Hartford, CT 06120, heidic@rh.edu

THE SOFTWARE ENGINEERING II COURSE

In this section, the contents of Software Engineering II and my instructional goals for SEII are described. Potential roadblocks to learning in the context of the SEII course are also identified in order to lay the foundation for understanding the techniques and approaches used to facilitate active, student-based learning described in the following section.

Contents

Software Engineering II, ECSE-6780, is a 15 week graduate topics course. The prerequisite to SEII is ECSE-6770 Software Engineering I (SEI) which is an introductory software engineering course intended for students who have had little exposure to software engineering. SEI provides students coming into SEII with an understanding of the basic elements of requirements, software architectures, design, testing, etc. to support the concepts of distributed object technology presented in SEII. In addition to the SEI prerequisite, it is also assumed that students are experienced with the object-oriented paradigm. SEII is taught once a year during the spring semester. SEII has an enrollment cap of 28 students, but enrollment has typically been lower with an enrollment of 11 students in the 1998 and 1999 semesters and 17 students in 2000. SEII is a topics course where topics are selected by the instructor. During the past three years (1998-2000), SEII has included an increasingly large component on distributed object technology. During the spring of 2000, topics included remote method invocation, the CORBA standard, and distributed components as well as issues related to distributed system performance, distributed database integration, distributed transaction processing, and scalability issues. It was difficult to find an academic text that supported a majority of these technologies. Based on feedback from students in the 1998 and 1999 classes, in 2000 I chose to use Robert Orfali's "Client/Server Programming with Java and CORBA" [4] and Dirk Slama's "Enterprise CORBA" [6].

Course Objectives

In this section, the course objectives for SEII are described as well as the learning goals of the instructor. Below are listed the course objectives for the spring 2000 semester:

- Understand the capabilities required of software that supports client/server and distributed computing
- Understand the role of Java in Web-based application development
- Understand the major communication mechanisms used in distributed object technology and how they operate
- Be able to evaluate and select among various communication and interaction mechanisms to support a distributed or client/server application
- Understand the benefits and drawbacks of object request brokers for distributed object technology
- Understand the use of the CORBA standard and its Services and Facilities
- Be able to develop an efficient application using distributed objects, Java, and CORBA

These goals define the framework of knowledge that should be conveyed during the semester.

Instructor Learning Goals

In addition to the stated course goals, I have identified three learning-based goals which I used to guide and direct the approaches taken to learning in the classroom. I started formulating these goals the first time that I taught SEII during the spring 1998 semester and I have continued to refine and focus these goals during the subsequent two offerings of the course. Given the constant change in the area of distributed object technology, my primary goal was to develop the ability of students to learn and think critically for themselves. Since the actual details of the technology taught in class will soon be obsolete, my main goal was to impart to students the ability to absorb, evaluate, and incorporate knowledge on their own, without depending on an outside authority (i.e., the instructor) to provide either the knowledge or the interpretation of the value of that knowledge.

A second major instructional goal in teaching SEII was to provide an opportunity for long-term learning to continue after the course had completed. I believe that the best way for students to attain long-range benefit from a course whose content is continually and rapidly changing is to initiate collaborative relationships among students and among students and the instructor that persist beyond the scheduled end of the course. These long-term relationships can provide the students with a knowledge network which allows students to both collaboratively learn from one another and teach each other long after the course in which the relationships were formed has ended.

My final teaching goal is directly related to my first two goals of independent critical thinking and lifelong learning. As a secondary goal, I desired to find a way to take advantage of existing student expertise in the middleware area to expand the learning experience of all students in the class while maximizing the flow of information between students. Since most students at RH are working in industry, a few of them come into SEII with some background in distributed object technology. My aim was to expand the knowledge base used in the classroom beyond my own knowledge and experience with a subject. I had also hoped that the sharing of knowledge would provide a means of initiating and establishing relationships among the students that would lead to the foundation of a knowledge network.

As I identified and pondered on the instructional goals for SEII, I realized that in order to achieve my goals, I had to shift the classroom paradigm from "instructor as expert" to "collaborative student learner". In the "instructor as expert" paradigm, students look to the instructor as the knowledge source and director of their learning. The instructor is the primary authority on a subject and provides the main core of knowledge as well as determining the importance of that knowledge and its incorporation into the students' larger body of computer science knowledge. In this model, most information flows in one direction, from instructor to students.

In the "collaborative student learner" model, with the aid of both the instructor and their classmates, students acquire knowledge based on personal motivation such as interest or job need. This learning is attained through student investigation where students go through a process of first determining and prioritizing the knowledge that is important to them, and then using various approaches to attain that knowledge including utilizing peer experience, literature and Web searches, experimentation, etc. In many cases, the information gained through personal investigation has greater impact and is remembered longer because of both the motivation behind the attainment of such information and the method used to acquire the information [7].

Challenges and Roadblocks

Teaching a topics course where the content of the course is continually mutating gives rise to a set of challenges that must be met for effective learning to take place. The instructional goals identified in the previous section bring an additional set of constraints to the learning environment. In this section, I identify the challenges and potential roadblocks to effective learning in the SEII course.

The first main challenge in teaching SEII is the difficulty of maintaining currency in the subject matter. Since the technology is being adapted and enhanced at an accelerated rate, it is difficult for the instructor to maintain fluency in the technology. The issue of currency became my major motivation for attempting to utilize some collaborative learning approaches in the classroom. I quickly recognized that I was not going to be able to maintain the "instructor as expert" paradigm.

The experience level of the student, or lack thereof presents the second major challenge in teaching SEII. Since the majority of RH's students work full-time, a few of the students entering SEII come into the course with knowledge of distributed object technology based on work experience. However, the majority of students entering SEII have little background in distributed object technology. This disparity of experience could lead to unequal learning experience among the students in the classroom. My challenge was to figure out a way to leverage the existing student knowledge without creating a class system between the students who knew the material and the students who did not.

A third challenge was related to the characteristics of RH students. Students at RH typically work full-time and may be geographically distributed, resulting in students having limited availability to the instructor and other students. In addition, I wanted to shift from the "instructor as expert" paradigm to the "collaborative student learner" model. My challenge was to create an environment that encouraged student cooperation and interaction, removing dependence on the instructor for learning and forming a climate of interdependency among the students. In order to meet this challenge, I needed to incorporate asynchronous, distributed communication mechanisms to support student interaction and collaborative learning among students who might only see each other face-to-face once a week.

COLLABORATIVE LEARNING IN SEII

In recent years, cooperative learning strategies have been gaining attention in the field of computer science education as a way to promote a more comprehensive learning experience for students. Chase and Okie [1] utilized peer instruction in a CS1 class to lower the failure and drop-out rate for beginning computer science students. Hagan and Sheard [2] incorporated discussion classes into an undergraduate microcomputer laboratory course in order to foster small-group learning and to clarify understanding of course topics. Hagan notes that when the discussion groups were removed, student learning was negatively impacted to such a degree that the discussion groups were reinstated. Yaeger et. al., [8] describes improved learning achieved in an undergraduate Dynamics course resulting from the use of collaborative learning.

Jenkins [3] describes a "learning through doing" approach to teaching an introductory computer science course. Jenkins reports more enthusiastic students that have a higher retention of course material. In addition, Jenkins noted that the class atmosphere is less formal, allowing students to more easily form relationships among themselves and with the instructor. A more extensive and inclusive approach to peer learning is presented by Rice et. al., [5] who reports on the creation of a peer learning community at the University of Pennsylvania.

In this section, I describe my general approach to incorporating elements of collaboration into SEII and discuss how I used this approach to address the potential barriers to effective learning. I then go on to describe the specific techniques used to support collaborative learning.

The General Approach to SEII

I begin this section with a discussion of the general approach that I took to teaching SEII. I first used this approach during the spring 1998 offering of SEII and have continued to distill and focus this approach during the 1999 and 2000 offerings. I previously stated that one of the major roadblocks to teaching SEII was the need to keep current in a subject area that is continually changing. This left me with a decision. I could either present a course based only on my somewhat limited knowledge, or I could attempt to provide a wider experience by relying on students to expand the common knowledge base for the class. I decided to take the latter approach and I turned the class over to the students. I constructed a general syllabus that listed the topics that I thought were of interest in the area of distributed object technology, but left the description broad enough to allow a wide choice of manners in which these topics could be covered. At the first class meeting, I told the students that I had collected a set of topics that I thought were of current interest and that we could study those topics if they were willing to embark on an atypical learning experience. I clearly stated that I could in no way be considered an expert in any of the subject areas, but that I was willing to provide some initial direction and that I would support a joint learning process. I indicated that the syllabus was very amendable and that I was willing to follow their learning objectives as long as I could incorporate those objectives into the academic framework of sufficient scholarly content and grading. I warned students that this approach to learning could be more frustrating as there would be no "expert" who would know all of the answers, but I also indicated that this approach would allow them to customize their learning. I indicated that I would never ask them to do something that I didn't know could be done, but that I would ask them to do things that I did not know how to do.

When the students agreed to try this new learning experience, a partnership was created. No longer was I steering the boat and the students pulling the oars. Now the students were navigating through the material and the learning process themselves with some general input from me. They were directing their own learning experience. The class and I made joint decisions about all exams and assignments. I suggested an example domain to which to apply the knowledge gained in SEII and we jointly developed assignments, spending cooperative class discussions defining the details of exactly what must be produced. Obviously I had the final say in all decisions, but I consciously tried not to exercise that power unless students were making decisions that were not feasible either from a scheduling or academic perspective. Note that I very rarely had to exercise the power of final decision.

Techniques for Collaborative Learning

The approach that I took to teaching SEII provided the foundation for the employment of several specific techniques for supporting collaborative learning. The fact that I stepped out of the role of the expert and simply lacked the knowledge to answer any and all student questions meant that students were forced to rely on themselves and their classmates for aid in problem solving, thus setting the stage for true collaboration to occur. In this section, I describe several techniques that I employed to foster a more comprehensive collaborative environment for students in the SEII course.

The first technique that I used was simply to remove any lecturing from the course. Instead, I split the three hours of class time between interactive class discussion and small-group labwork. I prepared few formal lecture notes and much time was spent drawing on a common blackboard and doing joint coding projects. Again, I turned responsibility for learning over to the students by having myself and the students jointly determine a set of questions that identified the holes in students' (and my own) knowledge or provided direction for student learning and then simply turned students loose to find the answers.

A second technique I used to foster conspirational learning was the joint development of a mental model of the subject matter. In order for students to understand a new domain of information, they must first construct a mental model of the domain which provides the framework understanding upon which additional learning can be built. Since distributed object technology is a relatively new subject area, while a few students had some knowledge of the subject, no one had a complete picture of the topic. I employed the approach of jointly building a mental model during class time. Each class was started by asking students to describe their understanding of the area of distributed object technology currently being investigated. This information was displayed on the chalkboard and students asked questions and added missing pieces to the model until everyone in the class had a relatively complete mental model. In some cases, alternative models were displayed by different students or groups of students and these models were compared and contrasted to increase student understanding of the topic. Once everyone in the class had a relatively complete understanding of the concepts, we then proceeded to code a small example that supported the piece of the model that we had added during discussion. This served to validate our mental model as well as to highlight any errors or omissions in our model. This model building process proceeded in an incremental fashion with an iterative "build the model-code-evaluate" cycle that we went through as many as ten times during the semester.

A second technique that I used with some success in SEII was brainstorming. After a mental model was built, students were presented with a problem that I knew could be solved, but one whose exact solution was not obvious. In class, students were required to create solutions to the problem either individually or in groups of two or three. These solutions were then jointly evaluated by all class members and students identified the benefits and drawbacks to the solutions as well as analyzing the effort required to effect the solution. One or more of the approaches were selected for trial and students worked together to code solutions. The solutions were then evaluated to determine how closely they fit the expected benefits and drawbacks and as well as how nearly they fit the effort estimation for the solution.

As part of building mental models and brainstorming, I structured class and lab time so as to maximize student interaction in a manner similar to the discussion classes described in [2]. I encouraged students to collaborate with their peers by directly asking students who had completed an in-class assignment to work with students who had not yet mastered the material. I selectively constructed small groups when doing problem solving and model building so that students were put into groups that had a continuum of experiences. I rotated people through the groups so that each class member interacted with all others. I attempted to have each student be a peer instructor [1] at least once during the semester. I sent selected students to research areas of interest to share with the class with no direct payback (i.e., no grade credit). Students were selected based on interest and completeness of mental model and results were shared either in class or via a listserv. Through the period of the course, I attempted to find an area for each student to research based on their strengths so that all students felt that they contributed to the body of the class model.

Perhaps the most important technique that I used was to simply lead by example. I let students know that I was not an expert in the area and that they all knew more about some aspect of the material than I did and that I would need their help in constructing a complete mental model. Then at the beginning of the course, I indicated the boundaries of my knowledge and we started to build a mental model. During this building process, I sat in a chair on the same side of the room as the students and asked them the questions that I needed answers to in order to understand the subject area. I formed relationships with my students with the goal of learning from them. I also turned all assignments into joint projects. Students were directly involved in determining the topic, content, and due date of assignments, and I completed assignments right alongside of my students. The general solutions to assignments were discussed in class, but students were encouraged to write their own code as many of them wanted to try new approaches. So not only did students learn collaboratively from each other, but I learned collaboratively from them as well.

In order to improve their critical thinking skills, when students asked me a question, I verbally walked through the solution procedure so that they could learn the process that I used to solve problems and acquire knowledge. I discussed how I would approach the problem, how I identified the key features of the problem and how I organized my thoughts towards finding a solution. I presented the solution directions that occurred to me and we opened the floor for discussion. Students quickly came to see that collaborating with their classmates increased the number of different possible solutions to a problem and improved their chances of finding a workable answer.

Communication is the key factor in supporting collaborative learning. Since the RH student population is geographically distributed, I made heavy use of asynchronous electronic forms of communication. I set up a listserv for the class and encouraged its use by posting answers to questions and code snippets. I made heavy use of a course Web page for posting assignments, links of interest, and pertinent documentation. I also used the Web site for dynamic problem development and solving.

OBSERVATIONS OF THE LEARNING PROCESS

During the process of implementing my approaches to creating and supporting a collaborative environment in the SEII course, I observed several interesting effects of the employment of the techniques that I used. This section describes my perceptions about the effectiveness of the techniques that I used in SEII to encourage and support collaborative learning and briefly summarizes how my learning goals were achieved.

After beginning the SEII class using the approach of turning much of the direction and control of the learning process over to the students, there was an immediate bonding between the students as they cooperated to decide the contents and approach to be used in the class. In the 1998 and 1999 semesters, students were familiar with each other (i.e., knew each other's names and jobs) and students had identified their classmates who had experience or knowledge in a particular area by the end of the second class meeting. Active collaboration occurred within three weeks where I would arrive in class to find students already asking and answering each other's question. This high degree of student interaction supported my goal of leveraging existing student knowledge in the subject area and provided the basis for long-term learning to occur.

Long-term learning is still occurring among students from the 1998 and 1999 semesters as evidenced by the fact that students from these semesters continue to communicate with each other. A core group of six students from the 1999 offering of SEII keep in regular email contact with each other, exchanging knowledge not only related to distributed object technology, but also pertaining to the students' job technology needs and other topics tangential to the SEII course. In addition, some students, mainly from the 1998 offering of SEII, use me to facilitate communication with other students. Students from the 1998 course email me with information that they ask me to pass along to other students. This continuing exchange of information indicates that knowledge networks have been constructed among at least a subset of the students and the goal of long-term learning has been achieved at some level.

Students were very enthusiastic about the self-directing approach that I used in SEII. Students from all three offerings of the course were excited about having the ability to decide the direction that their learning took and class participation was high. I had 100% attendance except in the rare cases where students were unable to attend class due to business commitments. In addition, class frequently continued for an hour or more beyond its scheduled end. Classes were often student run. Some days I would provide a start to the learning by beginning class with a set of general questions that I had come up with on the current topic. Other days, students would already be discussing a topic or problem when I arrived at class and their questions would guide class contents. This free-form approach to class meeting time allowed students to absorb knowledge from all of their classmates, achieving my goal of collaborative learning through the leveraging of existing student experience. The student-led approach also supported my goal of critical thinking as once students had been through the process of both brainstorming and collaboratively building a mental model, students gained the ability to both discover and solve problems on their own.

By the latter third of the semester, students were exhibiting a higher capacity to think critically about concepts and to apply newly gained knowledge to familiar environments and to assess the benefits and drawbacks of the application of that knowledge. This was evidenced by the fact that students became much more adept at both constructing mental models as well as accurately assessing those models as the semester progressed. Students transitioned from mental model to supporting code more quickly and with fewer errors. Brainstorming efforts became much more focused and effective. Students spent less time in brainstorming, producing more articulate, understandable solutions to problems in shorter time periods as compared to the beginning of the semester.

During all three offerings of SEII, I observed that students became more confident of their answers to questions and more sure of their thought processes in general as the semester progressed. This assurance provides evidence that my goal of honing student critical thinking skills was being achieved. I believe that students were more confident in the conclusions that they reached because the reasoning behind their answers came from their own analysis process rather than rationale provided by the instructor [7]. Throughout the semester, I noticed that students were more willing to take risks in learning, became correspondingly more secure in their ability to find solutions to their problems, and showed a greater willingness to forge collaborations with their peers. Students became more willing to be evaluative of their decisions and to learn from their mistakes as they increasingly based their decisions on self-knowledge rather than teacher-fed fact.

CONCLUSIONS

In this paper, I have described my experiences in teaching a collaborative advanced-level software engineering course, Software Engineering II. My experiences in teaching SEII have shown that taking a more student-directed, cooperative approach to learning can result in benefits in the areas of critical thinking, and long-term and collaborative learning for students. The more free-form approach to learning taken in SEII shifts the flow of information from instructor-to-student, to student-to-student, putting the instructor in the role of facilitator rather than expert. The final effect was much more highly motivated students as they had a direct stake in the material that they were learning. In addition, the student-led approach to learning used in SEII builds relationships that persist far beyond the boundaries of the classes in which the relationships were formed. The peer-based approach taken to SEII involves taking some risk. Using this approach, the instructor must relinquish much of the control over the course content and direction to the students. However, by putting the power (and responsibility) of learning into the students' hands, students gain a much more personal and useful learning experience.

In addition to the positive impact on students resulting from the student-directed cooperative approach to the SEII course, this teaching approach has raised some issues about future directions for the course. Over time I have built up a base of knowledge in the area. Current experiences teaching SEII have shown me that I must take care not fall into the "instructor as expert" paradigm which I believe will hinder collaborative learning. In addition, during the spring of 2000 I had 17 students which is the largest number of students I've ever had in SEII. I am currently ascertaining what changes need to be made to adapt the approach to work with larger groups.

In addition to the impact of instructor knowledge, my experience shows that learning needs some sort of a support system in order to sustain continued learning after the class has ended. Students still look to me as a facilitator for knowledge exchange if not for the actual knowledge. Since I cannot continue to facilitate communication for all past SEII students, communication among students needs to be supported without my presence, perhaps through some sort of a listserv or threaded discussion group.

Another question that has occurred to me as a result of my experiences in SEII is the query "Can collaboration be fostered across class boundaries?" I am currently exploring the feasibility of supporting cooperative learning across different offerings of SEII as well as the possibility of creating joint learning experiences between parallel offerings of SEI and SEII classes.

REFERENCES

[1] Chase, J. D., and E. G. Okie, "Combining Cooperative Learning and Peer Instruction in Introductory Computer Science," *Proc. of 31st Annual SIGCSE Conf.* March 2000.

[2] Hagan, D., and J. Sheard, "The Value of Discussion Classes for Teaching Introductory Programming," *Proc. of ITiCSE '98*, August 1998.

[3] Jenkins, T., "A Participative Approach to Teaching Programming," *Proc. of ITiCSE '98,* August 1998.

[4] Orfali, R. and D. Harkey, *Client/Server Programming with Java and CORBA*, 2nd ed., Wiley Computer Publishing, 1998.

[5] Rice, W., et. al., "New Pedagogical Approaches to Computer Science Education: A Case Study in Peer Learning," *Proc. of FIE '99*, October 1999.

[6] Slama, D., J. Garbis, and P. Russell, *Enterprise CORBA*, Prentice Hall, 1999.

[7] Sutherland T.E., and C.C. Bonwell, "Using Active Learning in College Classes: A Range of Options for Faculty," *New Directions for Teaching and Learning*, No. 67, 1996.

[8] Yaeger, P., et. al., "Interactive Dynamics: Effects of Student-Centered Activities on Learning," *Proc. of FIE '99*, October 1999.