New Orleans JAZZED about Engineering Education

Teaching Large Computer Science Classes

Dr. Shahriar Shamsian, University of Southern California

PhD degree in computer science from UCLA 13 years of teaching experience at UCLA and USC 25 years of experience in the software industry

Dr. Gisele Ragusa, University of Southern California

Gisele Ragusa is a Professor of Engineering Education at the University of Southern California. She conducts research on college transitions and retention of underrepresented students in engineering and also research about engineering global preparedness and engineering innovation. She also has research expertise in STEM K-12 and in STEM assessment. She chairs USC's STEM Consortium.

Prof. Jeffrey Miller, University of Southern California

Dr. Miller is an Associate Professor of Engineering Practice in the Computer Science Department at the University of Southern California. He earned his BS, MS, and Ph.D. from the University of Southern California in 2002, 2002, and 2007, respectively. He has taught collegiate Computer Science for over 10 years at California State University, Los Angeles, the University of Alaska Anchorage, and now at USC. His research in two discrete areas - Computer Science education for K12, undergraduate, and graduate students and intelligent transportation systems, specifically related to vehicular networking and ethics with driverless vehicles.

Dr. Michael Shindler, University of Southern California

Michael is a member of the teaching faculty at the Computer Science Department of the University of Southern California. He teaches a variety of classes covering programming, systems, and computer science theory.

Teaching Large Computer Science Classes

Dr. Shahriar Shamsian, University of Southern California

Shahriar is a member of teaching faculty at the Computer Science Department of the University of Southern California where he teaches graduate level courses in the area of design and analysis of algorithms. He has over twenty years of software industry experience and thirteen years of teaching experience at UCLA and USC, and has a PhD degree in Computer Science from UCLA.

Dr. Gisele Ragusa, University of Southern California

Gisele Ragusa is a Professor of Engineering Education at the University of Southern California. She conducts research on college transitions and retention of underrepresented students in engineering and also research about engineering global preparedness and engineering innovation. She also has research expertise in STEM K-12 and in STEM assessment. She chairs USC's STEM Consortium

Dr. Michael Shindler, University of Southern California

Michael is a member of teaching faculty at the Computer Science Department of the University of Southern California where he teaches a variety of classes covering programming, systems, and computer science theory.

Dr. Jeffrey Miller, University of Southern California

Jeff is an Associate Professor of Engineering Practice in the Computer Science Department at the University of Southern California. He earned his BS, MS, and Ph.D. from the University of Southern California in 2002, 2002, and 2007, respectively. He has taught collegiate Computer Science for over 10 years at California State University, Los Angeles, the University of Alaska Anchorage, and now at USC. His research is in two discrete areas – Computer Science education for K12, undergraduate, and graduate students and intelligent transportation systems, specifically related to vehicular networking and ethics with driverless vehicles.

abstract

Growing enrollment numbers in Computer Science programs in schools across the country are a reflection of the rapidly growing computer industry over the last few decades. Many schools have met the challenge of higher enrollment numbers by adding classes to address new course content and increasing the sizes of these classes. While the size of the more specialized classes may still be kept at a manageable and reasonable level, the core classes that most university students have to take present special challenges for the administration. Over the last ten years, we have, at different times, tried different approaches and used a variety of different class sizes to accommodate the higher enrollment numbers for such core classes.

Importantly, each approach has had its own advantages and challenges. In this paper, we present some of the techniques we have used that have helped us improve student learning in large class settings. We discuss challenges associated with large classes inside and outside of the class. We present this research as a case study of a particular large computer science graduate course with information that may be replicable to other large classes across computer science and engineering education. For our case study we selected a graduate level algorithms design class to demonstrate the effectiveness of different approaches to addressing the ever-increasing enrollment numbers for these classes. We share our experience with both pedagogical and logistical challenges in such class settings as these and present solutions for such challenges through a combination of technology and approach to teaching.

1. introduction

A primary goal of engineering education is to provide students with requisite technical grounding along with practice and experience in the design and evaluation of real and practical systems. This goal becomes increasingly difficult with the expanding body of knowledge, integration of concepts across disciplines, and complexity of design tools needed in engineering industries.[1] While an expert/apprentice model of education may be more fitting to preparing engineers for professional practice, traditional instruction models include in-person lectures covering fundamental technical concepts with the bulk of practice and application occurring outside of the classroom.[2] This comes typically in the form of homework and labs (possibly in the presence of a teaching assistant) with delayed feedback from e-mail, office hour meetings, and occasional graded assignments. Little time is available in class for modeling and demonstration of the desired practice skills by the instructor. This is particularly problematic for courses with large class sizes because large class sizes often prohibit significant individual modeling, assessment or interaction with students.[3]

The purpose of this paper is to share the approaches, techniques, strengths and challenges that we've had in trying to teach classes with high enrollment numbers. This paper is a "work in progress" paper in which we use a class a "case study" and describe both our evolution that we took in attempting to get the large course more engaging for students and to improve student learning. We discuss the evolution of our approach to teaching these classes both in terms of teaching methods used in the classroom and in terms of the supporting technologies. For our case study we use a graduate level class on design and analysis of algorithms as our class "case." The findings shared in this paper are based on over three years of performance data on the same course taught by the same instructor but employing different approaches to teaching. We believe that our experiences and findings in this study can be used and adapted to fit other course with large enrollment number.

This paper responds to two primary research questions:

- What are the challenges and potential solutions for engaging students in learning with large size computer sciences courses?
- What are effective pedagogical approaches to teach computer science in large classes?
- Which pedagogical approaches positively impact students' satisfaction and learning in large classes?

2. the course: design and analysis of algorithms

As previously described, findings in this paper come from our experience with a graduate level course on Design and Analysis of Algorithms. This course has consistently had highest registration levels within the Computer Science Department at our university because, not only is it a mandatory course for all computer science students in the Master's program, but because of its widespread application in other fields, it attracts many students from outside of the Computer Science Department at our university.

One of the unique aspects of this course is that for students to succeed in the class, they must not only understand the concepts, they must also be able to apply such concepts creatively in solving new problems. Creativity is a main ingredient in algorithm design. This aspect must be considered when designing the course material, exercise problems, and the overall approach to teaching this subject. For example, invoking creative thinking and creative reasoning processes should be a key objective of each classroom experience, and how that can be achieved in a small or large classroom setting will be different.

3. class size

Over the last several years we have tried several different strategies on class size, each with their own successes and challenges. We first look at some of the pros and cons from a teaching perspective in the course and then try to connect the various teaching methods to students' evaluations of the courses.

Following are the three approaches that we have used thus far. They are listed in the chronological order in which they were implemented. So, one could think of this sequence as the evolution of the pedagogical approach we have taken in teaching classes with large enrollment numbers.

3.1 approach #1: many small lecture sections

With this approach, class sizes were limited to approximately 50 students and various instructors were assigned to teaching one or more sections of the course each semester.

- The Advantages: An obvious advantage of such class settings is that they can be more interactive. Students can be more directly engaged in the creative thinking and reasoning processes throughout the session. They have substantive opportunities to ask questions and therefore have a better chance of following lectures more closely.
- The Challenges: Aside from staffing challenges in finding enough instructors for all class sections, a secondary challenge that had an impact on student's satisfaction from the course was inconsistency across the various sections. Even when the same syllabus was used across sections, the fact that some students could not register in the section taught by their favorite instructor was a source of dissatisfaction.

3.2 approach # 2: several large lecture sections

With this approach, classes were held in large auditoriums holding between 200 to 300 students in each session, with the same instructor teaching all lectures.

- The Advantages: This approach provided the much needed consistency in terms of syllabus, teaching style, exam difficulty, and workload across all class sections. Additionally, the staffing requirements were also easier to manage for the department.
- The Challenges: Not unlike mega section courses in universities nation-wide, this type of class setting provided students with very little chance to interact with the instructor in the class environment. Students had to wait for instructor office hours to ask questions or engage in discussions.

3.3 approach #3: combination of the above two approaches

Learning from past two traditional experiences with small and large lecture settings, we attempted to bring together the best of both worlds by combining large lecture sessions with small discussion sessions. In other words, the auditorium style lectures were augmented by discussion classes. Unlike discussion sessions that are taught by the more junior Teaching Assistants, it was decided that the discussion sessions would be taught by skilled faculty to allow for a high quality interactive learning environment. Additionally, groups engaged in problem solving with discussion sections that enabled immediate application of the particulars learned in the lectures.

- The Advantages: In addition to the advantages associated with the second approach stated above, i.e. providing consistency in terms of syllabus, teaching style, exam difficulty, and workload across all sections, plus the lower staffing requirements, this approach provided students with a venue in which they could fully engage with instructors, ask questions, and be involved in discussions with the instructor and other students.
- The Challenges: We have not yet experienced any major challenges with this approach now that we have engaged our seasoned faculty in discussion sections. This formula has so far proven to be the most promising.

The only additional effort involved in this approach as compared with the second approach is the required close coordination between instructors teaching lectures and discussion sessions so they can always stay in synch.

To determine the impact of the structural and pedagogical changes we have made to this large class, we studied students' end of the semester evaluations of the course to see if there is an obvious difference between the students' satisfaction of the course and the approach used in the teaching of that course. For the following analysis we are using the same instructor's records over a period of X semesters teaching the same design and analysis of algorithms course.

Table 1: Class Ratings by Semester and Section

	Approach I			Approach II	Approach III		
	Fall 2012	Spring 2013	Fall 2013	Spring 2014	Fall 2014	Spring 2015	Fall 2015
Section 1	4.64	4.39	4.36	4.17	3.97	4.43	4.38
Section 2	4.5	4.3	4.15	-	4.08	4.45	4.38
Section 3	4.14	-	-	-	-	-	-
Average per semester	4.42	4.34	4.25	4.17	4.02	4.44	4.38
Average for approach	4.34			4.17	4.28		

The above responses correspond to the survey question "Overall, how would you rate this course?" with following possible responses:

Table 2: Rating Schema

Rating	Description
1	Poor
2	Below Average
3	Average
4	Above Average

5	Excellent

Considering the range of course ratings for different sections taught in the same semester using the same approach (e.g., from 4.64 to 4.14 in Fall 2012), the difference between overall ratings for approach I versus approach III (4.34 vs 4.28) appear to be within statistical noise, implying that large auditorium style lectures on their own were not be perceived by students as undesirable, rather we believe that the key is the overall classroom experience independent of the size. In the following sections of this paper, we share some of the techniques specific to larger classes that we believe can contribute to a more successful classroom experience.

4. large lectures: challenges inside and outside of class

In this section, we discuss some of the challenges inherent in managing large classes either inside the class environment or when dealing of other logistical aspects of the managing the class. In the sections that follow, we will share approaches that have helped us deal with these challenges effectively.

<u>keeping students engaged in class</u>

One of the main challenges in providing a good learning experience in a large classroom environment is to keep students engaged with course content.[4] Techniques that work well in smaller classes such as getting students involved in problem solving by asking questions and discussion possible solution approaches can be problematic for example:

- Students are sometimes intimidated to speak up in a large auditorium.[5]
- Since students' questions may not be heard by all, questions either have to be repeated by the instructor, or the student has to reach for a microphone, in either case the interruption takes away from flow of the discussion.
- It is often practically not feasible for many students to get involved in discussions because of the sheer number of people in the class.

Importantly, although this form of student engagement may work at certain situations and for certain types of (very focused) questions, it clearly cannot be the only form of student engagement that the instructor can rely on.

<u>forming discussion groups</u>

Earlier is this paper, we also discussed the specific nature of the course we are studying for this paper, namely the fact that students need to learn how to apply their knowledge of the concepts to solving problems. For some students, the optimal environment for this type of learning is in groups where they can exchange ideas and discuss ways to tackle a problem.[6] A challenge in facilitating such a classroom environment in an auditorium at our institution is the fact that seats are fixed in certain positions and not suitable for forming groups.

<u>grading of exams</u>

Whether managing a small or a large class, it is widely acceptable that it is important that students' papers are graded fairly and consistently. However, providing consistency in grading of five hundred plus papers especially if grading involves assigning credit to partially correct answers in a reasonable period of time is a non-trivial challenge.

Since the grading process involves a number of resources that have to work in parallel to complete grading in a reasonable time, challenges involved are both quality related and logistical. For example, to maintain fairness and consistency in grading it is ideal to have the same resource grade the same exam problem for all students. But if work is to happen in parallel, to grade physical papers, each resource will have to work on a portion of the papers and pass them to the next resource when done. When there are five to seven resources involved in this process, the logistics of moving papers from resource to resource is itself a challenging task and can reduce the efficiency of the whole process. On the other hand, to simplify the logistics of moving papers around and to split papers amongst resources and to have one resource grade all questions for one student will result in significant inconsistencies in grading and can be a source of justified complaints by those unfairly graded.

5. keeping students engaged in class

As previously noted, engaging students in coursework in large classes is a significant challenge. In the sections that follow, we discuss some of the techniques that we believe have assisted us in improving students' learning experience in larger classes. To keep students involved and engaged in their learning, we have used a combination of techniques:

5.1 technique #1: posing "ponderable" questions

The ponderable technique is a mean's of posing questions to students that may not have one correct answer but enable to reflect on the content and its application.[7,8] This technique is used to allow students to have a chance to think about a question before the answer is given. Some of the parameters that can affect the success of this technique include:

- <u>amount of time provided to students to think about the question</u>
  Based on our experience, a period of at least 30 seconds to at most 3 minutes depending on the question type will have the best student impact. It is possible to lose students' attentions if they are given too much time on a given question.

- <u>Student discussion of the ponderable</u>
  Depending on the type of question, it may be beneficial for students to either think independently or to discuss and share ideas. More time should be given to those questions that require discussions since student's attentions are less likely to move away from the question if they are actively engaged in discussions.

- Frequency of use of the ponderables technique
  As any other technique, this cannot be overdone. And if this technique is used more than a couple of times in one lecture, it is best to vary the parameters to make each a different experience.

It may not be entirely possible to have these parameters optimized the first time a given question is posed in this format, however we have noticed that by gauging student's level of engagement in real time, these parameters can be optimized after a few trials.

5.2 technique #2: in-class exercise problems

As previously noted, the specific nature of the algorithms design course requires that students be able to apply their knowledge to solving problems. It is therefore necessary to get students involved in this aspect of learning during each lecture so they can practice under the guidance of the instructor.[7] There are a several parameters that should be considered when using in class exercises:

- the difficulty level
  Initially, it is important that students gain confidence in the ability to tackle new computer science problems, so it is beneficial to start out with easier in class problems. As student confidence is gained, it is also important to challenge students with more difficult problems, which can help build their interest and curiosity in the topic.[9]

- amount of time provided to "ponder" the problem
  The optimal time assigned to each problem may not necessarily correlate with the level of difficulty of the problem, because the intent behind the particular exercise may be different.[10] For example, the purpose of posing a simple problem during the lecture may be that students actually solve the complete problem, but with a more challenging problem, the intent may be that students only appreciate the complexity of the problem so they would be more interested in learning about its solution. Accordingly, the optimal amount of time assigned to an in-class exercise problem may depend on multiple parameters.

- discussion of problem solution ideas versus independent work
- Based on our experience, it is important to encourage teamwork on the more challenging exercise problems posed in class, so students have opportunities to hear different ideas and approaches to tackling problems before prior to seeing the solution.

5.3 technique # 3: in-class, real-time polling

Online polling systems provide an effective method of getting students engaged in large class environments.[11] We used this technique to manage question and answer sessions in the large class we are studying. This approach can be combined with the previous two techniques (i.e., "ponderables" and in-class exercises) and provides students with the ability to respond to questions or pick the correct solution and in that way stay more

engaged with the class. Students can visually see the incoming answers form a bar chart on the screen as the answers flow in and see their own answers contribute to results shown giving them another level of engagement in the happenings of the class.[3]

There are several benefits associated with the polling approach to collecting students' responses:

- It is anonymous. More students participate in providing responses if it is anonymous than if it weren't. For example, we have experienced that far fewer students raise their hands in response to a question such as "Who thinks this solution is correct?" than that in an anonymous poll.
- This interaction automatically engages the student with what is happening in class. Moreover, it does so in a different way than listening to the lecture.
- It provides students with a feel for where they are relative to the rest of the class in terms of their grasp of the topics and content discussed. Accordingly, it provides immediate feedback to them.

5.4 technique #4 using videos during lectures to support instruction

There is now a wealth of video content available online on almost every topic. Good quality and interesting videos, however that can actually help student learning in a graduate level course are hard to find. But if such videos are available, playing a short clip that enforces or reiterates the important points in the lecture has been shown to be beneficial.[12,13]

5.5 technique #5: forming discussion groups in lecture halls

Forming groups to work on problems or to discuss a topic provides students with an opportunity for active engagement. However, at times auditorium setting is not the most inviting for the formation of larger groups. While it may be possible to form groups of five or even more students to work together in smaller class settings, the nature of auditorium seating may limit group size to a maximum of three. When given the option to form groups of two or three, it has been observed that most students end up in groups of only two. Based on our experience, students often welcome such engagements, and even if there are only two students in the group, it provides them the occasion to interact which can help keep student's attention and interest in what is happening in the class.

5.6 technique #6: supporting technology

Some of the technology tools that have helped us in conducting and in general managing larger classes are:

- tools for grading papers
As mentioned, consistent and fair grading of a very large number of papers is a challenging task. To alleviate this, there are a number of tools that are currently available online that can help facilitate this process. The tool that we have been

employing over the last two years is CrowdMark. This tool allows us to upload scanned copies of exams to an online platform that provides where a number of resources can simultaneously access and grade all student papers. This tool also uses barcodes to identify exam papers which helps us perform all grading anonymously. The tool also keeps track of student scores at the question level, so adding up scores is done automatically and removes the possibility of human error in that process.

- tools for classroom content management
There are many options available to manage classroom content online, most often referred to as learning management systems (LMS). The tools we have used in the last ten years are Blackboard and Desire2Learn. These platforms both provide suitable environments to organize class content, post and collect homework assignments, post grades, send out announcements, etc. Our class sizes have been pushing the limit on some of the scalability aspects of these solutions, but overall, they have both been extremely useful tools in managing our large classes.

6. preliminary case study results

As previously described, tis paper represents a work in progress case study of a large lecture class. In comparing students grades across years since we have employed the new techniques and approaches described in this paper, we not only have seen increases in student satisfaction with the course, we have also seen increases in student learning as demonstrated by homework and course grades. We recognize that we are not comparing students to them selves grade-wise and therefore note that individual variation may account for some grade increases, however these preliminary results indicate a set of promising practices for addressing the unique needs of teaching large or "megs-section" courses in computer science. Our study is ongoing and therefore we have plans to expand the use of our approaches and techniques to additional courses within them computer science department and other engineering departments.

7. conclusion

We believe that our work in improving the learning experience for our students is an ongoing effort. The special aspect of this work related to handling of courses with large enrollment numbers is also part of this ongoing work. Having studied the results of the different approaches we have tried to manage such classes over the past several years, we believe that the current method of combining large lectures with small discussion sessions provides the best and most consistent learning experience for our students. We have used a number of techniques to help improve students' learning experience in large classroom settings. We believe that best results are produced by combining these approaches and techniques and adapting them to the subject at hand for each lecture. For example, an engaging format for a lecture might include a sequence of short segments:
- Lecturer speaking on a new topic
- Possibly followed by a short video with supporting/overlapping content
- This could be followed by posing a question/"ponderable"
- A poll can reveal students' responses on the question

- Lecturer continuing with the same or new topic
- Followed by students working together on a problem
- A poll on the solution results

Finally, managing very large classes is very time consuming and therefore it is important to have full support of the school and department to provide adequate resources such as instructors for discussion sessions, teaching assistants to work one on one with students on their issues, graders for the massive amounts of grading work involved, and a strong technical support team to help overcome the scalability issues that might be faced when using online classroom tools in class sizes for which they were not originally designed.

References

1. Heywood, J., (2005) *Engineering Education: Research and Development in Curriculum and Instruction.* John Wiley & Sons, Hoboken, NJ, 12-24
2. Kim, J., (2004) The Effects of a Constructivist Teaching Approach on Student Academic Achievement, Self-Concept, and Learning Strategies. *Asia Pacific Education Review* 6 (1), 7-19.
3. Mayer, R., (2004) Should There Be a Three-Strikes Rule Against Pure Discovery Learning?. *American Psycologist* 59 (1), 14-19.
4. Mayer, R. E., (2011) *Applying the Science of Learning* (Pearson Education, Boston, 197-209.
5. http://dailygenius.com/best-classroom-tools-gathering-feedback/
6. R. Clark and K. Yates, (2008) *Guided Experiential Learning: A Mentorship Perspective*, presented at Proceedings of the American Education Research Association Annual Meeting, New York.
7. National Research Council. (2012). Discipline-Based Education Research: Understanding and Improving Learning in Undergraduate Science and Engineering. S.R. Singer, N.R. Nielsen, and H.A. Schweingruber, Editors. *Committee on the Status, Contributions, and Future Directions of Discipline-Based Education Research. Board on Science Education, Division of Behavioral and Social Sciences and Education*. Washington, DC: The National Academies Press.
8. PCAST STEM Undergraduate Education Working Group, S.J. Gates Jr., J. Handelsman, G.P. Lepage, and C. Mirkin, Co-chairs. (2012). *Engage to Excel: Producing one million additional college graduates with degrees in science, technology, engineering, and mathematics.* President's Council of Advisors on Science and Technology.
9. Doğru, K., (2007) Applying the Subject 'Cell Through Constructivist Approach during Science Lessons and the Teacher's View. *Journal of Environmental & Science Education* 2 (1), 3-13.
10. Ragusa, G (2011) Best Practices in Pedagogy and Associated Assessment: Presentation to the National Science Foundation EEC Awardees. Arlington, VA.
11. Foertsch, J., Moses, G. Strikwerda, J. and Litzkow, M. (2002) Reversing theLecture/Homework Paradigm Using eTEACH® Web-based Streaming Video Software. *Journal of Engineering Education*, 267-274.
12. "Flipping the Biomedical Engineering Classroom: Implementation and Assessment in Medical Electronics Course", (2014) Lecture presented by Dr. Jean-Michel I. Maarek.
13. S. Zappe, R. Leicht, J. Messner, and T. Litzinger (2009) *"Flipping" the Classroom to Explore Active Learning in a Large Undergraduate Course*, presented at Proceedings of the Annual Conference of the American Society of Engineering Education.