I believe that teaching is an essential part of a meaningful academic career – if research is to have any significant impact on society, it must be shared with people at all levels, from students to colleagues, from fellow researchers to engineers. For me, teaching is a tremendously rewarding experience: not only does it allow me to share with students the knowledge and problem-solving skills they need to improve the world around them, it also helps me learn how to become a better researcher and educator.

1 Course development: CIS 505

I have taught CIS 505 (“Software Systems”) during each of the five semesters since I joined the tenure track. CIS 505 is a core systems course in the MSE program, and its final exam qualifies as one of the four WPE-I exams in the PhD program. The course has also recently become a core course in the EMBS program. Although it is a graduate-level course, it tends to attract a substantial number of undergraduate students; for instance, the class has had 17 undergraduates in the past semester and 18 undergraduates in the previous one.

CIS 505 teaches fundamental concepts of distributed systems and the design principles for building large-scale computational systems. The class covers key building blocks – such as synchronization primitives, group communication protocols, and replication techniques – that form the foundation of modern distributed systems, such as cloud-computing platforms or the Internet. Besides foundational concepts, it also covers real-world case studies of distributed systems, such as GFS, MapReduce, Spark, and Dynamo.

When I took over the course, I redesigned all of the materials: I developed a fresh set of slides, I designed a completely new set of homework assignments from scratch, and I created a new final project with a more ambitious scope. The first assignment is meant to familiarize the students with basic UNIX concepts and to give them some hands-on experience with synchronization and concurrency; this is useful because the course attracts a highly diverse group of students – including not only undergraduates, Master’s students, and PhD students in CIS, but also students from other departments – and some of them have never programmed in C/C++ before. The second assignment creates a set of email servers (SMTP and POP3) that work with a real e-mail client; this shows the students how a distributed system works under the hood, it gives them experience with socket programming and the client/server paradigm, and it provides opportunity to work with real-world specifications (RFCs). The third assignment asks the students to build a fully-distributed chat system with multiple servers that supports several different consistency models. This exposes the students to complex ordering protocols, as well as to practical issues such as packet loss and network delay, and the challenges of building a reliable system that can tolerate these issues.

The final project is to build a fault-tolerant cloud platform called PennCloud in teams of four. PennCloud is loosely inspired by Google Apps: the required features include a Gmail-like webmail service and a storage service that is somewhat analogous to Google Drive. This brings together many of the concepts that are discussed in class – such as fault-tolerance and consistency – and it exposes the students to the challenges of building a large software system in a team. The homework assignments are intended as building blocks and can be used as part of the project; thus, despite the limited time, the students can get a large system with a lot of functionality that they can be proud of, and that can later become part of their portfolio, e.g., for job interviews. The project includes a detailed proposal, work-in-progress presentations, a demo, and a final report, all of which provide an opportunity to practice important soft skills.

The feedback from the students has been very positive. The course and instructor ratings went up significantly and reached 3.1–3.5 out of 4 (with an upward trend). Based on the data in Penn Course Review, I was the first to achieve a rating above three for this course – despite the fact that the course is now more challenging, as reflected by the increased difficult level (previously 2.6, now 3.3). The course evaluations suggest that the students enjoy the course. Example comments include:

- “Wonderful course. Change nothing. Loved every minute of it”
- “The final project has led me to an offer from Google, thanks this course very much, learned a lot!”
- “Perfect balance of theoretical and practical experience. Leaves you with a feeling of extreme accomplishment when you are able to complete all the assignments and specially the project (which is by far the best project that I have done). Very hard, leads to a lot of sleepless nights but totally worth it. One of the best courses at Penn”
• “Best CIS instructor. Enthusiastic about her research and teaching. Always stayed after class and answered everyone’s questions. Assignment specs were detailed and clear. Piazza questions were answered timely and in great details.”
• “The course is amazing. I think it is a course that is fundamentally important in computer science discipline and would strongly recommend everyone to take this course.”
• “Professor Phan cares a lot about students getting something out of the course – she is a very enthusiastic, energetic, and approachable instructor.”
• “Never seen such a devoted instructor in my life.”
• “Hands down the best CS course I have ever taken at Penn.”

2 Mentoring

I believe that engaging students in research is a valuable teaching tool. In the past few years, I have actively involved undergraduates and Masters students in research – not only to provide them with experience in developing creative solutions for interesting problems, but also to encourage them to explore their potential and to pursue advanced degrees. Through my class, many students have developed deeper interests in distributed systems, and they have further pursued research projects with me on real-time cloud and NFV. For instance, I have been working with many undergraduate and master’s students (many of them are female students) through independent studies and senior design projects. As an advisor, I guide them towards understanding both the theoretical foundations and the challenges in designing and implementing practical systems, while also teaching them other aspects, such as literature survey and research writing. The projects often lead to successful publications; for instance, we turned Tianyang Chen’s master’s thesis into an RTAS 2018 paper that received the Outstanding Paper Award.

I have been fortunate to be able to advise several talented doctoral students. Three of my students already graduated: Jaewoo Lee (whom I co-advised with Insup Lee) is currently an assistant professor at Chung-Ang University in Korea, and Yang Li (whom I co-advised with Sanjeev Khanna and Boon Thau Loo) is now a researcher at Facebook. My third student, Meng Xu (whom I co-advised with Insup Lee) has recently graduated and started working at Apple. All three students had very solid track record – for instance, Meng had four RTAS papers, as well as papers at RTSS, ICCPS, EMSOFT, and CLOUD; one of his papers was nominated for the best paper award at RTSS 2013, and he is currently also working on a submission to RTAS 2019 and a real-time journal submission.

My other four students are on a great trajectory as well. Max Demoulin and Saeed Abedi, both of whom I co-advise with Boon Thau Loo, are at the beginning of their third year and are doing well. For instance, Max has already published two papers (ACSAS 2019 and SelfSDN 2018), and is currently working on a conference submission; he also won first place at the ACM Student Research Competition at SIGCOMM 2017. Neeraj Gandhi and Robert Gifford (who are not co-advised) are second-year and first-year students, respectively, and have already made substantial progress: Neeraj is submitting a paper to RTAS 2019, and is planning to submit another to SOSP 2019; Robert is already working with me on an extension of Meng’s earlier work, which we hope will lead to a conference submission soon.

3 Diversity and outreach

I believe that computer science educators have a crucial role to play in increasing the enrollment of underrepresented groups in computer science. I am committed to improving the representation of women in computer science and engineering, as well as to helping female students in their studies and research. At Penn, I have also participated in informal panels, sponsored by the Penn Graduate Women in Science and Engineering, to share my experiences with female graduate students, as well as giving talks, e.g., at WiCS High School Day for Girls. I have also actively (and successfully) recruited several female students as teaching assistants for my class.

Outside Penn, I have served in different capacities to foster diversity. I have also been serving as a mentor for female students in my community. As a member of the Diversity subcommittee within the IEEE Technical Committee on Real-Time Systems (TCRTS), I developed initiatives and policies that aim to increase transparency and to advance the representation of women and underrepresented groups within the real-time systems research community at large and in the leadership roles (such as PC members and PC chairs). Several policies that I co-developed have been implemented or are being considered by the IEEE TCRTS, such as keeping a large percentage of women and underrepresented researchers in the Technical Program Committee of RTSS and RTAS, no deadlines on weekends, childcare support at conferences for attendees with small children, travel support for female and underrepresented students, mentorship program for female students who attend the conferences for the first time, awards dedicated to women who made significant contributions to the real-time community, etc. Through our effort, we have begun to see changes in the
perception within the community and in the leadership; I plan to continue working on these issues to make long-term impactful changes to the real-time community.

As a member of the ACM Future of Computing Academy (FCA), I co-developed initiatives to support and foster the next generation of computing professionals from diverse backgrounds at large. For instance, I co-developed strategic plans for increasing diversity at multiple levels (early education, high education, and early career). I also believe that, beyond diversity in gender and race, diversity in research thinking and outlook is equally important. Together with other members in the FCA, I have been working on efforts to increase awareness, promote, and foster interdisciplinary research; our initiatives have been strongly supported by the ACM leadership.