COURSE IMPROVEMENTS AND PEDAGOGY

GEOFFREY CHALLEN

04/03/2017

My goal as an educator is to inspire students. But my responsibility as a scientist is to evaluate my teaching efforts with the same rigor as my group evaluates our research efforts. And my opportunity as a computer scientist is to apply my domain knowledge to creating more effective education tools. Today, keeping pace with technological advances is critical. Universities face a growing challenge from online education and massive open online courses (MOOCs), and must compete online while also continuing to control the price of a college degree. So just as my research group, blue, designs, builds and evaluates novel computer systems, as educators we are also designing, building, and evaluating novel approaches to education.

During my six years at UB I have focused on teaching core computer systems concepts and research principles to advanced undergraduates and graduate students. When teaching "CSE 421/521: Introduction to Operating Systems" I was challenged to determine how to transfer a highly-effective and extremely-challenging class from another institution but offer it with far fewer teaching resources. Out of this challenge emerged the experimental `ops-class.org` online instructional framework (§ **??**) and also the new idea of modular MOOCs (§**??**), a research direction I am currently pursuing in collaboration with Margo Seltzer from Harvard University.

With these experiences as a starting point, I designed and deployed our department's First-Year Seminar (FYS) offering on the Internet (§**??**) The Internet FYS provided a first opportunity to evaluate the success of a modular MOOC as one component of a hybrid online and in-person approach to a large course. I am incredibly excited about the opportunity to teach freshman students about the most transformative technology of our era. In addition, I am designing the course to both engage other educators and domain experts in contributing content as well as creating space for a variety of engaging in-class activities. My goal is to use this FYS as a recruiting tool to ensure that all UB students realize how exciting and relevant computer science is, and anticipate that this will help my department increase both enrollment and student excitement.

At UB from 2011–2017 my primary teaching responsibilities were a large flipped introductory course on the internet (CSE 199, §0.1), an introduction to operating systems (CSE 421/521, §0.2), and a graduate seminar on mobile systems (CSE 72x, §0.3). For two years I co-taught a graduate course on advanced computer systems (CSE 622). In the interest of space, this course is not described. **In all my courses I focus on giving students the opportunity to build and use computer systems.** While I enjoy lecturing, I know that challenging projects inspire them to learn far more outside the classroom than they can inside.

0.1 — CSE 199: How the Internet Works (`internet-class.org`)

In Fall 2016 I taught CSE 199, a new freshman course on "How the Internet Works". The internet is the most significant thing computer scientists have ever built. It's development is rich in history, design principles, exciting engineering, societal relevance, and deep human implications. I cannot imagine a more exciting topic or a better way to introduce students to computer science.

But my interest in developing the course is not only due to the material. The course is part of UB's new Freshman Seminar program, which is intended to allow freshman to interact with faculty in small groups of up to 25 students. However, the "seminar" that I am designing will be taught by multiple faculty to groups of 100 students. Swamped by growing interest, my department does not have enough faculty to offer 25-student seminars. Instead, we used a combination of online learning and flipped classroom activities to provide personalized learning. My challenge was to use technology to scale the small-group seminar experience to large groups.

In Fall 2016 I supervised four teaching faculty and 25 undergraduate TAs in offering this course for the first time. All primary instruction happened online. Content is broken into 292 short videos of roughly five minutes, with approximately 100 minutes of content assigned each week. The entire playlist totaled 292 videos over 15 weeks. However, in contrast with many other online courses using this approach, eventually I will not be the only author. Instead, we are building `internet-class.org` to allow anyone with knowledge about the internet to contribute. By making it easy to add content and invite others to do so, I anticipate that we will eventually have many academic and industry contributors. Students were also required to provide explanations as a course assignment, and we plan to use strong student submissions during subsequent offerings.

Multiple explanations for each concept allow `internet-class.org` to adapt to students. Choosing the best content and reinforcements for students will help them learn the material. If my explanation is best, then students may see it first, but be offered multiple other explanations by other faculty and contributors as needed. In other cases, we may be able to learn what types of explanations work well for each student and steer them in the correct direction. This approach utilizes what I call learning webs and it forms the cornerstone of the educational plan in my funded CAREER proposal. Modularity inherent to our approach also allows other faculty to rearrange content to suit their own courses on the internet.

The flipped approach freed classroom time for new activities. Together with the course staff, I led the development of 24 new in-class activities on topics related to the internet. We aimed for a variety reflecting the diversity of the course material. Some activities were scavenger hunts. Others had students using command line UNIX networking utilities. Others had students playing online (security) games, or using existing online tutorials (codecademy). Others had students learning about parallel processing using decks of cards or routing using toothpicks and marshmallows.

0.2 — CSE 421/521: Introduction to Operating Systems (`ops-class.org`)

I'm a professor because I took a course on operating systems using the OS/161 instructional operating system from Margo Seltzer in 2001. While serving as a TA for the course six times as an undergraduate and Ph.D. student, I repeatedly witnessed the transformative effect it had on students. I was determined to bring this course with me to UB to train new generations of systems programmers. But this effort immediately ran into a major hurdle. At Harvard, five experienced TAs would be assigned to a class of 30 students (1:6). At UB I was provided two inexperienced TAs for 100 students (1:50).

I worked for several years to adapt the course for UB and continue to improve it incrementally. The results are visible at `ops-class.org` where class lecture videos and slides, assignments, exams, an online forum, and testing resources are available. To help students learn, I videotape all my lectures and post them on YouTube. As of July 2016 my lectures have been viewed 177,120 times and watched for a total of 1,343,806 minutes by 1901 subscribers. This allows students to review the material and prepare for exams. My lectures are also being viewed by people across the world, increasing the visibility of UB's CSE program.

My course staff and I have also made major advances in grading automation for the OS/161 assignments. Automated grading was introduced in the first year that I taught the course. Last spring, we improved the process further through a new `test161` tool. `test161` represents a novel form of distributed automated grading. Secret injection during compilation secures testing output from manipulation by students. This allows us to share tests with students and let them test on their own machines, shifting load away from our servers. Final submissions are made through the `test161.ops-class.org` website. An interactive web interface allows students to watch online grading. Scores are posted on public leader boards with student permission, encouraging friendly competition. An equally important benefit of automated grading is allowing students to improve their answers. Students may submit as many answers as they want until the deadline and use testing results to iteratively improve their submissions.

Automating grading has achieved its goal—allowing TAs to spend more time helping students. In Spring 2015, my TAs and volunteers held 27 TA-hours of office hours per week—approximately the same number per student as at Harvard. It is also more support than any other course in my department.

Despite both being extremely challenging and required for undergraduates, my OS course consistently ranks as one of the top CSE courses. Students have affectionately nicknamed it the "Challen challenge." Note also while other CSE classes struggle with low course evaluation participation, my course is evaluated by almost 100% of students each year. I accomplish this by releasing selected final exam questions early once the class meets course evaluation participation targets. I consider this a reasonable tradeoff. The course benefits each year from students' feedback, and high participation rates ensure that it is representative. And statistics show that students perform no better on the questions released early.

The success of my OS course has led students into my research group and to continued involvement with the course. In Spring 2014, we began a volunteer TA program. Three previous 421/521 students volunteered to serve as course "ninjas" and hold several office hours a week. Based on the success of that program we expanded it to 10 volunteers in 2015, 11 in 2016, 13 in 2017, primarily undergraduates from prior years. The willingness of previous students to volunteer speaks strongly to the course's success.

0.3 — CSE 72x

Since 2013 each fall I have taught a small seminar on a current topic in mobile systems. Course titles have included "Smartphone Sustainability" (2013), "Personal Cloud Computing" (2014), "Using Uncertainty to Program Mobile Systems" (2015), and " Improving Smartphone Quality of Experience" (2016). My seminar offers me an opportunity to work with students that are considering a Ph.D. Several previous enrollees, including Ali Ben Ali and Scott Haseley, have joined my research group. Due to the novelty of these topics, few readings are available that directly address the subject matter. However, this allows us to explore research in related areas such as sensor networking, OS design, energy management, and mobile systems. Class time is used for discussions. Randomly selected students summarize the paper and lead discussion, requiring all students to read and understand all assigned papers. During the course of each semester I usually assign implementation projects related to the specific course content. As the culmination of the course students work together to write a paper summarizing our findings. Papers produced by previous seminars have appeared twice at HotMobile (2014, 2015), the top workshop on mobile computing.