

Teaching Statement

Michael Ekstrand

Computer scientists are familiar with requirements. They establish what a program should do and may help us understand how it fits into the broader world. In software engineering classes, we teach our students to identify and document the requirements for a program or system, and let these requirements drive the design; even lightweight methodologies are driven by documented needs.

I find this concept to also be a useful guide to my teaching. For each course, assignment, lecture, etc., what are its learning objectives? What does it need to accomplish? Good software design and good education design have a lot in common.

At a high level, when designing a course, I first try to identify the purposes the course serves. Why is it in the curriculum? If other courses list it as a prerequisite, do they expect students to have certain skills or knowledge as a result of having taken it? What do students need, or want, to get out of it? If I am going to certify, with a grade, that a student has learned the relevant material, what is necessary to earn that mark?

The research project structure for the *Recommender Systems* class I taught at Texas State (CS 5369Q) is one example of this thinking in practice. One high-level goal of this course was to teach my students to engage with research literature. Their research engagement culminated in the final project: a report surveying 4-6 research papers and proposing an experiment that would extend or clarify the results of the surveyed papers. However, many of my students had no experience reading research literature; to help them learn to read and process research, I assigned a paper each week and had them write a short reading report identifying the paper's goals or research questions, the authors' methods, and the key findings. They could use 2 of these papers in their final survey, and we also discussed a number of the papers in class. By providing many low-stakes reports, giving direct feedback, and working the things I saw in the reports into subsequent classes, I could help students refine their ability to read and understand research. We also did the first reading report as an in-class exercise, so that they could have a model to work from through the rest of the semester. Overall, the project was a success: students' reports improved over the course of the semester, they turned in a number of good surveys, and some thanked me for the assignments.

My teaching experience is wide-ranging, from early-stage programming (data structures at UMN) and undergraduate electives (databases and UNIX programming) to graduate topical courses (Recommender Systems, both at Texas State and UMN). At the graduate level, I can teach human-computer interaction courses, including collaborative and social computing, as well as machine learning or AI courses as needed. At the undergraduate level, I would be able to teach many different courses; my primary interests are in user experience design, web programming, and programming languages, but have the background to teach many different courses depending on the department's needs.

In addition to traditional classroom experience, I also have experience designing and teaching a MOOC. With my adviser Joe Konstan, I co-teach *Introduction to Recommender Systems*, offered via Coursera. The course's first offering had 28,000 students registered, and we have since re-opened it as a self-paced course with 47,000 registrations to date. We worked with instructional designers from the university's academic technology to design the course, establish learning objectives for the class and its constituent modules and assignments, and develop meaningful learning activities and assignments, and collaborated with university technology researchers to study the effectiveness of the course (the results of this study are published in ACM TOCHI).

I think MOOCs are an interesting experiment in education that we don't yet know how best to use and deploy. My primary teaching goal is to provide an excellent education to a wide range of students; I am excited to see what we learn in the coming years about how best to use traditional classrooms, small- and large-scale online education, and other educational techniques to that end.

In my time at Texas State and my Ph.D studies at Minnesota, I have also worked with and mentored a number of undergraduate, masters', and junior Ph.D students. In particular, many of them have contributed to the LensKit project, where I have supervised their programming work and helped them to conduct research with it, write papers describing the results, and apply for research funding. My first masters' students at Texas State are currently undertaking their thesis research, and will hopefully have publishable results by next semester. Requirements and learning objectives play a role in my relationships with them as well: understanding what their goals are for their research experience and working to find projects and a path to get them from their current skill and experience level to where they want to be.

I have been programming for many years, love the craft, and believe that computers can be a powerful force for good in our society. I hope to encourage my students to pursue excellence as they harness computing to make the world a better place.