

Teaching Statement: Zachary G. Ives

September 29, 2008

Academia provides an opportunity to combine research and teaching in a way not possible in any other setting: to teach classes, get students engaged in research, and ultimately feed some aspects of that research back into the classroom. I attempt to take my teaching responsibilities to heart, both by considering how to take advantage of the teaching/research interaction, and also by continually re-evaluating how trends in industry should reflect upon my teaching.

Teaching Philosophy

My teaching philosophy acknowledges that Computer Science as an academic subject must serve at least three audiences with dramatically different goals: some students study the discipline purely for its own sake; others want to learn Computer Science to help support some other pursuit (e.g., animation, biology, etc.); and still others hope to pursue a professional technology career (e.g., programmer or technician). As an example, my CIS 550 “Database Management Systems” course, which averages 45-50 students every year, has members of all three audiences: a few students are undergraduates who hope to go to graduate school to study Computer Science in depth; about half the class are Master’s students who want to become software developers; and the remainder are students whose primary interest lies in supporting finance, telecommunications, genetics, computational biology, and so on. When teaching such a diverse class, I believe that it is vital to continually motivate why the topics are important to *all* of the students – this is especially true with technically dense or mathematical topics that some students may find intimidating. I try to emphasize how a deeper understanding may ultimately help the students tune or debug an application, keep their data consistent, connect to the Web, etc.

I am also a strong believer in keeping students engaged through a “semi-structured” and interactive class experience. From the very first day, I set an expectation that I will be asking questions, and initiating discussions that require active participation. Like many of my colleagues, I use PowerPoint slides to help organize and summarize lecture material. However, my slides have portions that are deliberately left blank and must be filled in during class discussions, or worked out on the board with student input. Students must come to class, reflect on the presentation, and participate in the discussion to get the most out of the material.

Finally, I try to get students *excited* about the topics I teach, through two primary means. First, it can be very inspiring to the students to see presentations from their peers – or from my PhD students – about ongoing research here in the department. I try to have guest lectures from students where appropriate. The sense of being connected to the leading edge can stimulate interest not only in the course material, but ultimately in getting involved in independent research projects. Finally, even course assignments can be made less rote and more interesting if they allow for creative freedom. I try to encourage and reward creativity and experimentation, through extra credit challenges, trivia questions, and so on.

Curricular Innovations

Computer Science is in a transition period as a field: after the “dot-com bubble” and the subsequent period of job outsourcing, there has been a dramatic drop in student interest in Computer Science careers. Yet, at the same time, there has been incredible growth and innovation in the field, particularly in Internet-centric and Web-centric technologies, sites, and services that people use every day. Virtually everyone interacts with Google services on a daily basis; families keep in touch using photo sharing sites like Flickr or Internet telephony and videoconferencing applications like Skype; social networking sites such as Facebook have taken the world by storm; YouTube and iTunes are revolutionizing the audio and video realm. Text messaging, iPhones, and Blackberries have changed communication forever.

I believe that our teaching of Computer Science needs to be updated in accordance with this network-centric, communication-centric, data-centric world. As a first step in this direction, I have created a course called CIS 455/555, Internet and Web Systems: this course is intended as a “capstone” to our advanced undergraduates and Master’s students, and one that would help launch them towards careers in Web-related areas. My course combines topics from distributed systems, information retrieval, Web services, and data interchange, with a focus on issues related to scalability, reliability, consistency, and distribution. As the course project, the students work in teams to build an academic clone of Google. In my course evaluations, I have received multiple comments from students saying this was their favorite course in the CIS curriculum – and the most useful.

The department is now in discussions to generalize some of these ideas, and to restructure some of our curriculum to form an entire undergraduate (and perhaps Master’s) concentration on what my colleagues and I are terming “Networked Information.” I have been an active participant in these discussions, and plan to assist in the creation of a new sophomore-level Web/cluster-centric programming course and a revised distributed systems course.

Courses

In my time at Penn I have created and/or revised three different courses.

CIS 550, Database Management Systems. I significantly “modernized” this course, which was based on the Ramakrishnan/Gehrke book and which covered mainstream topics in database design, modeling, and internals. I added a substantial amount of new material on XML and its languages; converting between XML and relations; data integration; and query processing and its impacts on tuning. I feel that these topics are not only timely and good “resume builders” for MS and undergraduate students, but they also have many connections to existing research going on in the Database Group at Penn. The students in this course (around 45 on average) are often from fields outside of computer science (e.g., genetics and computational biology, the Wharton School, etc.), so I strive to balance between practical and research topics. I continually stress how the material – whether theoretical, systems-oriented, or applications-oriented – is useful to the students, whatever their background and goals.

CSE 455/CIS 555, Internet and Web Systems. Penn has a reputation for teaching computer science with a very theoretical and algorithmic flavor, and not for teaching practical programming skills. To help address this, I designed CSE 455/CIS 555 to be a “capstone” course for Penn’s undergraduate and Master’s students, with an emphasis towards *practical* systems-building principles, viewed from a cross-cutting perspective. The course synthesizes material from databases (database back-ends, declarative data management, XML and integration) with distributed systems (scalability, partitioning, remote procedure calls, peer-to-peer protocols, time synchronization, etc.) and information retrieval (text search, ranking). The course teaches large-scale development and teamwork skills (the final project is to build a highly scalable peer-to-peer Web crawler and search engine, with ranking and the ability to interoperate with other engines). I strive to establish connections between the topics in lecture and real-world artifacts (e.g., Akamai, Google, DNS, etc.) and to familiarize students with state-of-the-art tools as well as the principles underlying them.

CIS 650, Building Data Management Systems. I designed this course (to be offered every 2 years under the new designation of CIS 652) as a follow-up to CIS 550, oriented towards PhD students and Master’s students interested in doing research. CIS 650 focuses on principles and techniques for constructing data management, storage, and integration systems. I cover traditional database techniques, such as concurrency control, indexing, and recovery, as well as more modern topics such as XML query processing, data integration, peer-to-peer computing, and search engines. Since the role of this course is primarily to train PhD students, I emphasize both writing and system-building skills in this course: the students must turn in frequent paper summaries, as well as a project proposal that I grade for style as well as content; and the final project must include a substantial implementation and evaluation.

Summary

I believe the role of a professor in a Research One university is to be an educator informed by the “real world” and its needs, good pedagogical techniques, and the best ideas from research. I strive to do this, and to encourage and engage students both in the classroom and in the less-structured setting of research.