While my primary emphasis of my career so far has been research, I greatly enjoy teaching and the classroom experience. I have actively sought opportunities to teach, both while completing my doctorate at Georgia Tech and during my postdoctoral fellowship at New Mexico. I'm also fortunate to have an ongoing opportunity to mentor students at New Mexico, as I assist Barney Maccabe and Patrick Bridges with the direction of the Scalable Systems Lab. The feedback I have received from these efforts, both from faculty and students, has been very encouraging. My goal is a tenure-track assistant professorship in which I can perform research, teach graduate and undergraduate classes, and work closely with students.

Philosophy

My experience as a professional technologist and software developer has most directly informed my attitudes as a teacher. I believe that understanding a technology comes most directly from hands-on experience, learning not only what it does but why it does what it does. The important corollary here is that experimentation is not only desirable, but necessary for true understanding. Things that work but are not elegant, or those that do not work, present important opportunities for iterative design and evaluation. This approach can enhance the coverage of historical material as well as hands-on projects. Perhaps most important here is the opportunity to involve others in the iterative process through group projects, term-long projects that build on themselves, and class interaction through such activities as code reviews.

One of the most effective ways of demonstrating the relevance of classroom topics is to show how they translate into improved tools for use by computer scientists (or, increasingly, by non-computer scientists). Stressing the historical roots of "new" technologies demystifies them and encourages students to begin to think critically about them. This is the first step toward experimenting with technology, which is in turn the first step toward learning it.

Communication and interaction are essential at all points. At least as important as the production of some classroom artifact (such as a programming project) is the ability to communicate about that artifact: What does it do? Why? Does it fulfill its requirements? How? Instructors must provide examples and feedback to guide students in developing these communication skills.

One of the most rewarding things about learning is that non-trivial effort is required. If that effort needed to master a skill or understand a concept is acknowledged up front, the sense of accomplishment and self-confidence realized by the student after going through the process is that much greater. This is not to say that teachers cannot make learning fun, or that learning isn't enjoyable in and of itself. However, I do believe that a recognition that work is involved pays off in the end.

It is much easier to teach a motivated student. I believe it is easiest and most beneficial to motivate students by helping them understand the *relevance* of their classwork. Many undergraduates approach classroom topics with a single question: "How will this help me get a job?" This kind of attitude can easily result in an adversarial relationship between instructor and student. In my experience, an approach that emphasizes that there is work involved, but also emphasizes the relevance of that work, provides a framework where students are willing to invest time and effort in their own education. This in turn provides the instructor with several and varied opportunities for teaching *actively*: frequently interacting with students, encouraging critical thinking and analysis by students, and making the class a participatory experience rather than a passive one.

Experience

My experience in teaching Operating Systems to undergraduates at both the University of New Mexico and Georgia Tech has also been a significant influence. In both instances, I was responsible for preparing the class syllabus and reading list; delivering all lectures; creating all homeworks, programming projects and exams; and supervising teaching assistants or graders. My students were junior- and senior-level undergraduates, as well as graduate students from other disciplines. I find it productive to view the class material as an evolution of

concepts, and so organized the class content to emphasize the connections between those concepts. For example, my lectures presented the development of Remote Procedure Calls as a logical consequence of the adoption of both programming abstraction and networked computing. I then demonstrated how these basic ideas persist today in the current realization of RPCs in technologies such as SOAP and .NET.

I have also had numerous opportunities to act as a mentor, both in industry as the leader of a team of developers and in academia with graduate students. I feel very strongly that it is impossible to be a good academic researcher without also being a good mentor. The scope of work necessary to perform quality research demands the assistance of quality graduate students. Retaining those students, ensuring that they make progress in their own careers, and helping them deal with the work/life issues endemic to graduate school requires a strong mentoring relationship. Teaching also provides an opportunity for mentoring students — while presenting research papers as a lecturer, I have made a point of discussing what makes a research paper good or not and how quality research is conducted and communicated.

Goals

My research expertise is in operating systems, distributed computing, computer security, and high-performance and ubiquitous computing applications. I am interested in teaching courses on these topics at both the graduate and undergraduate levels. I am also capable of teaching undergraduate classes in several other areas (such as networking, programming languages, and software engineering). I would also welcome the opportunity to teach seminar-style courses that explore the state-of-the-art in current research.