Teaching Performance Indicators

Abhik Roychoudhury
School of Computing, National University of Singapore
abhik@comp.nus.edu.sg

1 Designing new modules

I have proposed and designed one new graduate module called *Automated Software Validation* which covers different verification methods (model checking, theorem proving) for software validation. In addition, I have also designed the entire course contents of an undergraduate module (*Critical Systems and their Verification*) which part of the Computer Engineering programme at School of Computing. *Module folders for both of these courses are included in my teaching portfolio.* Following are some further details about these two courses.

- *CS 4271 Critical Systems and their Verification*
  It is a relatively new course offered by School of Computing. It is only one of the few electives currently offered for the Computer Engineering students. I was solely responsible for designing the entire course (curriculum, assessment scheme, lecture material etc). Since this course is associated with the Computer Engineering program, I made this course a hands-on course. During the lecture/consultation hours, I show the students some hand-on tricks in modeling and verification through some online exercises (e.g. during the lecture itself we work out the modeling and verification of a substantial real-life protocol). Also, about one-third of the grade is based on performance of the students in a practical term project. The term project is done in groups of 2-3 people (bigger groups are required to take on bigger projects). A typical project involves formally specifying and verifying a substantial part of a real-life embedded system (e.g. a protocol or a hardware unit). To train the students appropriately for the project, I use the following schedule.
  - 1st month: I present a comprehensive list of potential projects to the students. Projects are fixed for each student.
  - 2nd month: Students finish the formal specification work and present an interim project report.
  - 3rd month: Most of the experimental work on formal verification is done; final submission.
- *CS 5219/6214: Automated Software Validation*
  This is another new course offered by School of Computing for graduate students. The purpose is expose students to sophisticated software debugging and program understanding techniques which go beyond traditional simulation and testing. I have

designed the course to cover a wide variety of validation techniques which include various static checking techniques (theorem proving, model checking and static analysis). The techniques are discussed and explained in a concrete setting with lots of emphasis on a term project. Since this is a graduate course, the students here are required to perform the projects individually.

2 Non-traditional Curriculum Development Activities

The reputation of a department or university rests not only on its research output, but also what the future employers say about the students we produce. With this in mind, since 2002 I have been actively involved in formulating a new written PhD Qualifier Examination in Computer Science at NUS School of Computing. This involved (a) deciding on set of core topics for examination (b) deciding on type and level of difficulty of questions and (c) setting sample question papers. In addition, since a formal Qualifying Exam is new to NUS School of Computing, the entire process of setting up the written Qualifier Examination was an intense activity. It involved consultation/discussion with many colleagues regarding (a) why a written Qualifier Exam is needed, (b) what is the desired format/level of difficulty of the exam and (c) assessment policies and administration of the exam. Our written Qualifier Exam consists of two papers — one in Theory and the other in Computer Systems.

Since the inception of the written Qualifier Exam in 2005, I have **served as the coordinator of the Theory paper in the Qualifier Examination**. The exam is offered twice a year. In each offering of the exam, my duties involve meeting students to clarify their queries about the exam, finding examiners to set questions and moderate the level of difficulty of the questions. I maintain a web-page for the Computer Science Theory Ph.D. qualifier which provides more details about the exam — `http://www.comp.nus.edu.sg/~abhik/CS-QE`

3 Student Supervision

While supervising undergraduate and graduate students, I have given equal emphasis to developing technical excellence and developing strong communication skills. This has involved long hours sitting through many practice talks and going through many re-writings of technical papers with my students. This has sometimes been achieved by delaying the writing of technical publications e.g. I have on one occasion made a graduate student rewrite a proof five times without writing it up myself only for the sake of making the student familiar with writing proofs. In other words, my student supervision is *not* driven just by the goal of making a student productive in research/publications. Instead, I strongly believe in giving *well-rounded training* to my students.

So far, one of my Ph.D. students (Xianfeng Li) has graduated and joined the Computer Science Department of Beijing University, the top university in China. **Two of my Ph.D. students have been awarded the prestigious Microsoft Research Asia fellowship** for their research work (Tao Wang — 2004-05, Vivy Suhendra — 2006-07). This fellowship is awarded to PhD students all over Asia and Australia on a competitive basis.

4 Pedagogical Article on Teaching

Apart from classroom teaching and student supervision, I have participated in workshops with educators from other universities to reflect on my experiences from teaching. *A summary of some of my teaching methods appears in a recent paper* entitled "Introducing Model Checking to Undergraduates". This paper was presented at the Formal Methods Education Workshop 2006 at Toronto. The paper can be accessed from the following website

`http://www.comp.nus.edu.sg/~abhik/pdf/fm-ed06.pdf`

It has also been included in my teaching portfolio.

5 Use of novel software tools for teaching

I have used software tools developed in my research for teaching undergraduate courses. The Chronos software performance estimation tool was developed as part of my research in 2002 - 2004. I am currently using it for laboratory assignments in the undergraduate course *Hardware Software Co-design*, an elective in our Computer Engineering programme.

6 Participation in Development Programmes

In 2001, I completed the Professional Development Programme (PDP) offered by the Center for Development in Teaching and Learning (CDTL) to incoming NUS teaching staff.