Teaching Statement

Boon Thau Loo
http://www.cis.upenn.edu/ boonloo/

As a faculty member, I place equal emphasis on teaching and research. I believe that teaching and research have a synergistic relationship. My education goal mirrors my inter-disciplinary approach towards research: I aim to ensure that students develop a holistic view of distributed systems by drawing stronger connections between the networking field, and related areas in data management, formal methods, and programming languages. Consequently, I have developed inter-disciplinary doctoral seminar courses exploring topics at the interaction of databases, networking, formal methods and programming languages.

I design courses that have a significant experimental component, where students work in teams to develop sizable software systems based on concepts learned in class. These include building an operating system, fault tolerant distributed mail/chat server, and a peer-to-peer search engine. I have introduced several new courses and improved two existing core classes in software systems and operating systems. Several of these courses are not only core requirements for undergraduates and graduate students, but they are among the highest enrolled courses in our Computer and Information Science (CIS) department.

Post-tenure, I have also embarked on new teaching initiatives that enhance our industry engagements. These include conducting preceptorial classes to Wharton/Engineering students to promote entrepreneurship, organizing Penn Software-defined Networking (SDN) hackathon teams to participate in competitions organized by Juniper Networks and Comcast, and inviting industry guest speakers from Brocade, Comcast, and Juniper Networks to my seminar classes.

1 Experimental Systems Courses

I will briefly highlight three experimental systems-oriented courses that I have taught at Penn.

1.1 CIS 380: Operating systems (Fall 2011-2013)

CIS 380 (Operating Systems) [12] is a required core class in operating systems (OS) for our CIS undergraduates. Historically, the OS course had an accompanying half-credit lab section, but this separate lab was removed in 2010 from the curriculum. Consequently, when I took over this class in Fall 2011, I conducted a major restructuring of the course material, so that the core OS implementation concepts previously covered in the lab could be integrated into a single full credit course with many more students participating in the implementation projects.

Together with one doctoral student and four undergraduate teaching assistants (TA), I created a new project sequence called *PennOS*, in which students develop a full-fledged operating system in user-space. PennOS aims to introduce students to hands-on experience in actually building an OS that exposes students to a wide range of OS concepts, including kernel-/user-level separation, process life-cycle, priority scheduling, signaling, file-systems, and shell development. PennOS runs on commodity PCs with no changes to the underlying operating system or need for installing/running virtual machines or use of specialized software. This makes it easy for system administrators to support the course even with large class sizes, or when there are multiple project courses within the same department.

PennOS requires students to program in groups of four, working together to develop a complex system with approximately 6000 lines of code on average. To get started, students have to read through substantial documentation and develop their own test code for learning. Students build their OS from scratch, and for many students, this constitutes the most substantial implementation project in their undergraduate studies. As a result, it is important to encourage and teach strong software engineering concepts. This includes requiring students to make use of code repositories for collaborative development, and to develop components separately with future integration in mind.

My TAs and I have written an ACM SIGOPS Operating Systems Review (OSR) journal paper on PennOS [1]. Student feedback on PennOS has been overwhelmingly positive. While the class is rated as one of the toughest CIS course, in the course evaluation, students found the project “very rewarding”, “very interesting and intellectually challenging”, and felt the projects made them “more comfortable around low-level C than ever before”. They also reported that the project “was an incredible experience” and “hoped to have more classes like this one". A majority of students gave the class the highest possible score of 4 (on a scale from 0 to 4) on the amount learned in this class in terms of knowledge, concepts, skills and thinking ability.

1.2 CIS 505: Software Systems (Spring 2008-2011, 2015-2016)

CIS 505 (Software systems) [16] is geared towards advanced undergraduates and first-year graduate students, and is one of the core classes that graduate students have to take as part of their graduation requirements.

The course touches on various aspects of distributed systems (e.g. OS concepts, clocks, elections, distributed mutual exclusion, fault tolerance, etc.). Since taking over the class in Spring 2008, I have added contemporary topics related to cloud computing and data center technologies. I have also made significant changes to the final project. Students work in teams to build a mail client/server based on the POP3 and SMTP RFC specifications, and then extend their implementations to enable fault tolerance and distribution. The project provides students with the experience of implementing systems from RFCs, and extending their implementations based on concepts learned in class, specifically the use of decentralized total ordering, elections, replication, and vector clocks. The course teaches teamwork within a team (of 4 students), but also *across* teams, where students have to get their distributed mail servers to interoperate with the servers and clients of other teams, to form a federated mail ecosystem live on demo day.

As indication of the course's popularity, the enrollment increased by 173%, from 38 students in Spring 2007, to 104 students in Spring 2011. My course/instructor ratings were also the highest since 2003 (since course evaluations were made available online). The head TA for the course (Rafi Rubin) received the outstanding teaching assistant award for 2008-2009. In the recent two offerings of the course (Spring 2015 and 2016), I have overhauled the final project to evolve the distributed mail server project into a distributed chat server, keeping many of the technical elements from the earlier project in place. The class enrollment has grown to 149 (Spring 2016).

1.3 CIS 553: Networked Systems (Fall 2007-2010, Spring 2012-2013)

In Fall 2007, I started an advanced undergraduate / graduate level networking course titled *CIS 553 (Networked Systems)* [10]. This course has been offered annually for the past 5 years. In this course, students learn about various networking concepts related to the Internet, wireless networking, and protocols across various layers (from link-layer to application-layer). Students work in teams to develop *PennSearch*, a substantial networked systems programming project (>10000 lines of code) using network simulator 3 (ns-3), an emerging open-source network simulator that is aimed at replacing the popular ns-2 simulator. In PennSearch, students develop various components in layers throughout the semester: first developing two protocols for Internet Protocol (IP) routing, followed by a distributed hash table (DHT) overlay network, and finally, a keyword-based search engine. One novelty of our assignments is the use of ns-3 in a large class setting, where students navigate through hundreds of thousands of lines of existing code before adding their extensions. In addition, in Fall 2010, selected groups developed the final project using declarative networking, a novel declarative framework that allows protocols to be rapidly synthesized using a high-level logic language.

A paper describing PennSearch was presented at the SIGCOMM Education workshop [4]. To the best of our knowledge, we are one of the first (in 2010) to attempt using the ns-3 platform for course projects that span network and application-layer protocols in a large class setting. Since Fall 2008, this course has been cross-listed with Penn's Telecommunications and Networking (TCOM) program offered by our Electrical and Systems Engineering department (replacing TCOM 512), reflecting its cross-departmental appeal. The course enrollment has grown steadily, from 34 students in Spring 2008, to 106 in Spring 2012 and 96 in 2013.

Student feedback from the use of ns-3 has been overwhelmingly positive. When the TCOM program was active, this course has consistently been ranked as one of the most popular TCOM courses (both in terms of highest enrollment and teaching ratings). During the period when I taught the class, it was also one of the largest (in terms of enrollment) 500-level elective course in the CIS department.

2 Inter-disciplinary Courses

I lead the NetDB@Penn research group, which explores research that involves building distributed systems that are provably safe and secure, combining techniques from databases, networking, programming languages, and formal methods. Given the inter-disciplinary nature of my research (see my research statement for more details), I have also created three research seminars that aim to bring together students from various sub-disciplines of computer science, and to explore cutting-edge research at the boundaries of various topics. I briefly describe them below.

2.1 CIS 700: Networking-meets-Databases (Spring 2007)

In Spring 2007, I created a new seminar course titled *Networking-meets-Databases* [11] which introduces students to current research at the intersection of databases and networking. The class was attended by several doctoral students in diverse areas of networking, databases, and security. As indication of its success, in its first offering, papers that emerged from final projects led by students have been published in various networking, security, and database workshops, with a number of follow-up conference paper submissions. The seminar also led to co-advising two students (Yun Mao and Micah Sherr) and planted initial seeds that led to two Ph.D. dissertations. Micah Sherr's dissertation received the Rubinoff award for most outstanding dissertation research in our CIS department.

2.2 CIS 800: Rigorous Protocol Engineering (Fall 2011)

Together with Penn post-doc Dr. Alex Gurney and Dr. Limin Jia from CMU, I created a new seminar course titled *Rigorous Protocol Engineering* [13] in Fall 2011 that introduces students to current research on the rigorous design, implementation, and analysis of network protocols. This course is in part inspired by Princeton's *Formal Methods in Networking* [3] seminar taught in Spring 2010, in which I participated as a co-instructor. The topics covered in this course include formal mathematical models of Internet protocols, domain specific languages (declarative, functional, or logic-based) for specifying protocols with high-assurance, and verification techniques to prove correctness and security properties of protocols. Case studies are drawn from traditional IP protocols, BGP policy configurations, secure BGP, transport protocols, application-layer overlay networks, and protocols developed using emerging platforms such as OpenFlow. Through these case studies, students acquire knowledge in formal modeling of protocols, and apply well-known verification techniques such as model checking and automated theorem proving into the analysis of network protocols. Industry speakers from Telcordia and AT&T Labs Research were invited to give students practical perspectives on the use of verification tools in telecommunications.

2.3 CIS 800: Software-Defined Data Centers (Fall 2015)

In Fall 2015, I created a new seminar course titled *Software-defined Data Centers* [15], a hot topic in both academia and industry. Software-defined data centers (SDDCs) refer to a data center where all infrastructure is virtualized and delivered as a service. The seminar explores the building blocks behind SDDCs, in particular, recent innovations in Software-defined Networking (SDN), Network Functions Virtualization (NFV), and virtualization technologies for storage and compute. The class covers topics spanning data centers, cloud computing, programming languages, and big data platforms. In the seminar class, not only do students read research papers from top networking and systems conferences, I also invited 5 guest speakers from Brocade, Comcast, and Juniper Networks, to give live demonstrations and brainstorm practical use cases of the research with students. The class is also closely integrated with my research, where we carry out a user study of the NetEgg tool for programming SDN policies by examples. As evidence of the class's success, the class received ratings of 3.77/4.

3 Mini-courses in Programming (2010 - 2013)

In Spring 2010 - 2013, I organized and administered for the CIS department's popular mini-courses, a series of half-credit CIS 19X programming courses (in C++, C#, Python, and Unix/Linux) offered to undergraduates. The courses are taught once a week, and each lecture is conducted by graduate students or qualified undergraduates. These mini-courses offer a win-win situation, where graduate students get to hone and cultivate their teaching skills, while undergraduates benefit immensely from practical skills obtained from these courses. In the past two years, I have achieved the following milestones in the mini-course series:

- The enrollment in these courses has increased significantly, often times with positive course/instructor reviews. Prior to Spring 2010, these courses each typically had an enrollment of 20-30 undergraduates. Today, the Unix/Linux and Python courses have an enrollment of 91 and 50 respectively in Spring 2012, the highest for any mini-course in recent history.

- New mini-courses in Haskell, iPhone programming, and Ruby on Rails are introduced. These courses are immensely popular among our students. I have also actively engaged the Philadelphia business and technical community, by hiring entrepreneurs Dr. Jim Young and Mat Schaffer to teach the iPhone and Ruby on Rails classes, respectively.

- Due to increased interests and the growth of these courses, mini-courses are now offered every semester (as opposed to only in the Spring semester). Mini-course instructors who are doctoral students are allowed to earn teaching practicum credits via teaching the mini-courses.

- Adam Aviv (Python course instructor) received the Center for Teaching and Learning (CTL) fellowship, a highly competitive university-level fellowship given only to a small group of graduate students each year. The Unix/Linux instructors Seth Shannin, Kevin Xu, C++ instructor Zach Zarrow, and iPhone TA Ayaka Nonaka received undergraduate departmental awards (John Brainerd, Hugo Otto Wolf, Cwilka, Exceptional service, respectively) for their contributions to the mini-courses.

4 Wharton Courses and Promoting Penn Entrepreneurship

As part of my teaching engagements, I also actively promote entrepreneurship among Penn students in engineering and Wharton. I am a serial entrepreneur myself, having started two companies, one of which is based out of commercialization of my research work. I regularly share my experiences on entrepreneurship with students, and provide them connections to potential investors and customers. For example, I conducted two invited lectures (preceptorial courses) in Spring 2015 and Fall 2016 to a well-attended audience consisting of students from Penn engineering and Wharton business school, sharing my experiences as an entrepreneur. I gave a guest lecture organized by Penn's Center for Innovation (PC) on university technology transfer.

During the period of 2008-2013, I have served a student mentor for the *NUS Overseas College - BioValley* (NCBV) [9], an international undergraduate exchange program administered by Penn Engineering and the National University of Singapore (NUS). Overseas exchange students enroll in classes at Penn while working as interns at a high-tech startup in the Philadelphia vicinity. As a mentor, I regularly meet and mentor undergraduate exchange students from Singapore, and also visit them during their internships at local startup companies. During this period, I mentored over 10 students, some of whom have gone on to start their own companies. In the years to come, drawing upon my cultural background and ties to Asia, I hope to play an active role at Penn, in promoting our interactions with universities in Asia.

Finally, I currently serve as the faculty advisor to Penn's M&T Innovation fund, a student-managed fund that provides pre-seed funding to startups by Penn students. This program not only provides funding, but also valuable experience to our students in evaluating potential startups and providing marketing/technical expertise to early stage companies.

5 Student Development through Teaching

In the past few years at Penn, I have actively involved undergraduates and Masters students in research, and have emphasized the participation of women and undergraduates in my research projects. Please see my research statement for more details. In this section, I focus primarily on student development activities related to coursework.

I passionately believe that computer science educators have a crucial role to play in increasing the enrollment of underrepresented groups in computer science. I encourage female computer science students to apply to be TAs, and several female students have served as TAs in my systems-oriented courses. These female students serve as inspiration to other female students in class, and a majority of them are our undergraduates and masters students. This is especially critical in systems-oriented courses such as CIS 380 and CIS 505, which involves large systems projects. Female TA highlights include Karen Tao (CIS 553 Fall 2009), Qiong Fei (CIS 553 Fall 2010), Sandy Clark (CIS 505 Spring 2009), Zhuoyao Zhang (CIS 505 Spring 2011) Yiqing Ren (TA for CIS 553 in Spring 2012), Ayaka Nonaka (iPhone mini-course instructor), Jennifer Cahalane (Ruby on Rails mini-course instructor), Connie Ho (CIS 380 Fall 2012/2013), Allison Pierce (CIS 380 Fall 2013), Charu Jangid (CIS 380 Fall 2013), Wenyi Wu (CIS 505 Spring 2015), Yifeng Zhu (CIS 505 Spring 2015), Saira Hussein (CIS 505 Spring 2015), Aayushi Dwivedi (CIS 505 Spring 2016), Bingying Liu (CIS 505 Spring 2016), and Yang Cao (CIS 505 Spring 2016).

I have also involved a large number of undergraduates over the years on various independent studies, in order to encourage them to think critically beyond coursework material, and possibly pursue a Ph.D. in computer science. Some highlights include:

- Mihai Oprea received 3rd prize for the senior design project competition in 2009. His work contributed directly to the RapidNet declarative networking system, demonstrated at SIGCOMM 2009 [8].
- Bill Marczak's senior design project in 2009 resulted in two first author publications in competitive database conferences (CIDR [7] and SIGMOD [6]). His work has also made significant commercial impact, through collaborations with LogicBlox. He is currently a Ph.D. student at the University of California Berkeley.
- Andrew Mao's senior design project in 2010 contributed to a publication in NDSS 2010 [14]. He is currently a Ph.D. student at Harvard University.
- Yash Saini's and Aditi Jain's senior design project in 2011 received honorable mention for the senior design project competition, and contributed towards the PUMA [5] demonstration at COMSNETS 2012.
- Samuel Appelbaum, Geoffroy Balbin, Evan Mossop, and Matthew Vogel received honorable mention in 2012 for their project on a bluetooth-based platform for delay tolerant networking on Android phones. Their system is available as an open-source library [2].
- Greyson Gregory, Vince Mannino, and Alex Marple received 3rd prize in 2012 for their project on a peer-to-peer MapReduce infrastructure over the Berkeley Open Infrastructure for Network Computing (BOINC).
- Perk Lun Lim, Albert Kwon, Alex Zhang, and Martin Pan received 1st prize for senior design project competition in 2013 for their design of the ROTOrouter router that can defend and mitigate security attacks. The team received honorable mention at the SEAS level competition. Albert is now pursing Ph.D. studies at MIT.
- Chenyang Lei, Sudarshan Muralidhar, Doron Shapiro, and Michelle Socher received 1st price for the senior design project competition in 2016 for their design of a secure distributed file sharing platform.
- Constanza Figuerola, Samy Lanka, Utkarsh Shah, and Max Tromanhauser received 2nd place in 2016 for their automated parking system that uses computer vision for license plate recognition.
- Charles Cobb, Meyer Kizner, and Xiuruo Zhang received honorable mention in 2016 for their bluetooth-based mobile messaging system.

References

[1] Aviv, A. J., Mannino, V., Owlarn, T., Shannin, S., Xu, K., and Loo, B. T. Experiences in Teaching an Educational User-Level Operating Systems Implementation Project. ACM SIGOPS Operating Systems Review (OSR), 2012.

[2] DroidDTN. `http://code.google.com/p/droiddtn-framework/`.

[3] Formal Methods in Networking. Princeton University. `http://www.cs.princeton.edu/courses/archive/spring10/cos598D/FormalMethodsNetworkingOutline.html`.

[4] Gill, H., Saeed, T., Fei, Q., Zhang, Z., and Loo, B. T. An Open-source and Declarative Approach Towards Teaching Large-scale Networked Systems Programming. In *SIGCOMM Education Workshop* (2011).

[5] Liu, C., Correa, R., Gill, H., Gill, T., Li, X., Muthukumar, S., Saeed, T., Loo, B. T., and Basu, P. PUMA: Policy-based Unified Multi-radio Architecture for Agile Mesh Networking. In *4th International Conference on Communication Systems and Networks (COMSNETS) (paper and demonstration)* (2012).

[6] Marczak, W. R., Huang, S. S., Bravenboer, M., Sherr, M., Loo, B. T., and Aref, M. SecureBlox: Customizable Secure Distributed Data Processing. In *SIGMOD* (2010).

[7] Marczak, W. R., Zook, D., Zhou, W., Aref, M., and Loo, B. T. Declarative Reconfigurable Trust Management. In *Proceedings of Conference on Innovative Data Systems Research (CIDR)* (2009).

[8] Muthukumar, S. C., Li, X., Liu, C., Kopena, J. B., Oprea, M., and Loo, B. T. Declarative toolkit for rapid network protocol simulation and experimentation. In *SIGCOMM (demo)* (2009).

[9] National University of Singapore Overseas Colleges - Bio Valley (NCBV). `http://overseas.nus.edu.sg/colleges_bioValley_intro.htm/`.

[10] Networked Systems. `http://www.cis.upenn.edu/~boonloo/cis553/`.

[11] Networking Meets Databases. `http://www.cis.upenn.edu/~boonloo/cis700-sp07/`.

[12] Operating Systems. `http://www.cis.upenn.edu/~cis380/`.

[13] Rigorous Protocol Engineering. `http://netdb.cis.upenn.edu/cis800-fa11/`.

[14] Sherr, M., Mao, A., Marczak, W. R., Zhou, W., Loo, B. T., and Blaze, M. A3: An Extensible Platform for Application-Aware Anonymity. In *Network and Distributed System Security* (2010).

[15] Software-defined Data Centers. `http://netdb.cis.upenn.edu/cis800-fa15/`.

[16] Software Systems. `http://www.cis.upenn.edu/~boonloo/cis505/`.