

University of Washington
May 31, 2007

Program Self-Study Report

Bachelor of Science in Computer Engineering

A.	BACKGROUND INFORMATION.....	4
1	Degree Titles.....	4
2	Program Modes.....	4
3	Actions to Correct Previous Shortcomings.....	5
4	Contact Information.....	6
5	CSE Department Web Sites.....	7
B.	ACCREDITATION SUMMARY	8
1	Students	8
2	Program Educational Objectives	15
3	Program Outcomes and Assessment.....	24
4	Professional Component	53
5	Faculty.....	60
6	Facilities.....	63
7	Institutional Support	65
8	Computer Engineering Program Criteria	65
	APPENDIX I	67
A.	TABULAR DATA FOR PROGRAM.....	67
	Table I-1. Basic-Level Curriculum - Computer Engineering	68
	Table I-2. Course and Section Size Summary	70
	Table I-3. Faculty Workload Summary	71
	Table I-4. Faculty Analysis	71
	Table I-5. Support Expenditures.....	71
B.	COURSE SYLLABI	71
B.1	Required Core Courses	71

B.2 Specified Courses in Hardware and Software Tracks.....	71
B.3 Hardware and Software Capstone Courses.....	71
B.4 Other CSE Majors Courses.....	71
C. FACULTY CURRICULUM VITAE.....	71
D. SUPPLEMENTARY MATERIAL.....	71

A. Background Information

1 Degree Titles

The Department of Computer Science and Engineering offers a **Bachelor of Science in Computer Engineering** degree through the College of Engineering. The program has two specializations, namely, hardware and software, which were instituted in the fall of 1999.

The Department also offers a **Bachelor of Science in Computer Science** degree through the College of Arts & Sciences. Although our students are differentiated by the degree program they choose, they are otherwise treated very similarly with respect to admission and advising, and in the classroom. We find that the mingling of computer science and computer engineering students in our classes has been very beneficial to both programs. Our computer engineering students end up with a very strong background in computer science, and our computer science students are exposed to engineering practice and skills.

2 Program Modes

Almost all undergraduate students in the department are enrolled full-time on-campus, although some students do attend part-time. Courses are offered during day-time hours all four quarters of the academic year (with a minimal summer quarter). Students must meet a satisfactory progress criterion that requires them to complete three courses towards graduation every quarter they are registered (except summer). Exceptions are granted for cooperative work experiences and extenuating circumstances like medical leaves, and for students nearing graduation with minimal outstanding courses to complete.

Most students enter our program in their sophomore year or even the start of their junior year. This is the case for historical reasons to guarantee articulation of transfer students from the state's Community Colleges and to ensure students successfully complete prerequisites prior to matriculation in the department. However, it has the negative effect of pushing all the requirements for our program into approximately 6-8 quarters rather than the 12 for which students are typically enrolled at the University.

In 1999, we started admitting incoming freshmen directly to our major. This was a new program initiated by our department to attract strong in-state students who may choose to go elsewhere due to the uncertainty of admission to the major. Now, both in-state and out-of-state students are eligible to be admitted through direct admission. Having students admitted early also means they can distribute their major requirements through their four years on campus, have more even workloads and possibly more time to engage in research. This has become a University-wide program and we (as well as many other departments) are now admitting 20-25% of our majors directly from high school.

3 Actions to Correct Previous Shortcomings

The previous ABET review found a very strong program, with no weaknesses or deficiencies. However, four concerns were listed.

Criterion 2 – The department should be better at exchanging information about the program objectives and soliciting of suggestions for improvement from constituencies.

We responded to this concern by initiating a discussion of the program objectives to establish a new set of objectives that were more in line with the ABET definition. This discussion included our industrial affiliates, faculty and students. At our annual affiliates meeting, we had a special session at which we proposed a number of possible objectives, including current objectives to industry representatives. We then held a similar discussion in a faculty meeting. A small committee incorporated the comments and feedback of the affiliates and faculty to formulate a set of program objectives, which were then published on our Website for further comment. We also presented these objectives to students for their comments. This resulted in the final set of objectives presented in this report. We will continue to review the objectives with these constituencies on an annual basis.

Criterion 3 – We should strengthen our measurements of the degree of achievement for each particular program outcome and provide better documentation.

We have taken several steps to strengthen our assessment efforts and provide better documentation. These are described in detail in Section B.3, but include the following improvements:

We have associated each outcome with a set of courses where that outcome can be assessed via directed assignments, exam questions and project assignments, or in-class presentations and discussions. Each course now has a small number of outcomes that are assessed directly by the instructor. Instructors must describe the assessment methodology for each outcome for that class, and include the assignment or exam question if appropriate, along with samples of student work and an assessment of each student in the class of their mastery of that outcome. These assessment materials are then collected and collated in a central place that allows them to be easily reviewed by course coordinators and the program coordinator. As part of this assessment effort, we have provided a set of assessment rubrics for some of the outcomes such as ability to work in teams and ability to communicate. These give instructors a set of specific criteria for judging these outcomes, and provide a more standardized assessment of these outcomes by faculty.

We have introduced an end-of-program review that is conducted by the Center for Instructional Development and Research (CIDR) of students in the senior capstone courses. These interviews specifically focus on our objectives and outcomes as they relate to the ABET criteria. These discussions in a town-hall setting allow the CIDR staff to probe more deeply than surveys. The results, which are a combination of student comments, consensus opinions, and quantitative self-evaluation on a range of outcomes, are reported to the department. The Undergraduate Program Coordinator reports these to the faculty and action is taken to address issues that are highlighted.

As mentioned above, we have started holding a special session at the annual affiliates meeting on educational issues where we discuss with our affiliate members our

educational goals, and ask for their assessment of our students and where we need to make improvements.

The College has introduced specific ABET-related questions on the student course evaluation forms. We will be tracking this data as it relates to specific computer engineering courses.

Criterion 4 – The incorporation of realistic constraints into the capstone design projects should be increased, and the evaluation of student competency in the communication and satisfaction of realistic constraints should be enhanced.

We have introduced several new capstone courses for the software option that cover a wide range of software systems. These involve complex, state-of-the-art projects in diverse areas like Internet systems, robotics, computational biology, operating systems and tablet computing. Designing these systems inherently involves consideration of a range of realistic constraints. However, we understand the need for students to articulate their consideration of constraints in the design process and in particular their need to consider tradeoffs when choosing among alternative solutions. The faculty teaching the capstone courses are asked to explicitly require students to incorporate a discussion of how constraints were met in the development of their design, and assess their ability to do so effectively.

Criterion 6 – A new building is necessary to assure adequate space and facilities.

We are pleased to report that we moved into a spacious new building in Fall 2003, with greatly increased laboratory and meeting spaces. This new space is described in detail in Section B.6.

4 Contact Information

Department Chair:

Henry Levy
104 Paul Allen Center
University of Washington
Seattle, WA 98195
(206) 543-9204

Undergraduate Program Coordinator (and lead author of this document):

Carl Ebeling
590 Paul Allen Center
University of Washington
Seattle, WA 98195
(206) 543-9342

Associate Chair for Education:

Richard Anderson
582 Paul Allen Center
University of Washington
Seattle, WA 98195
(206) 543-4305

5 CSE Department Web Sites

Main CSE Web page	www.cs.washington.edu
CSE ABET page with Objectives & Outcomes	www.cs.washington.edu/education/ABET/
Computer Engineering program requirements	www.cs.washington.edu/education/ugrad/current/New_CE_reqs.pdf
Information on all CSE educational programs	www.cs.washington.edu/education/
Information for current students	www.cs.washington.edu/education/ugrad/current/
Undergraduate Handbook	www.cs.washington.edu/education/ugrad-brochure/handbook.html
Course Home pages	www.cs.washington.edu/education/course-webs.html
Faculty Home pages	www.cs.washington.edu/people/faculty/
Undergraduate program brochure	www.cs.washington.edu/education/ugrad/prospective/ugrad2007.pdf
Recent capstone videos	www.cs.washington.edu/info/videos/
Computing and laboratory facilities	www.cs.washington.edu/lab/

B. Accreditation Summary

1 Students

The department operates an Undergraduate Advising Office with three full-time professional advisors (Crystal Eney, lead advisor; Megan Reardon, advisor; Raven Avery, advisor) and one part-time advisor, Monica Duke. The Undergraduate Program Coordinator (Carl Ebeling) supervises the overall advising effort. The office provides both pre-major advising to prospective students and advising and curriculum planning to current majors. The CSE Handbook can be found online at <http://www.cs.washington.edu/education/ugrad-brochure/handbook.html>. It summarizes in detail the program requirements and advising process. There is a separate Graduate Advising Office supervised by another faculty coordinator for our two graduate programs.

The CSE Advising Office supervises both the computer engineering and computer science majors. All advisors are knowledgeable about both degree programs and can advise any CSE student. The program information for the two majors is closely coordinated to take advantage of the commonality of the two degrees. Since the students from the computer engineering and computer science programs often take the same courses, many of the numbers and statistics presented in this self-study necessarily reflect combined data from both programs.

Admissions

The Undergraduate Advising Office coordinates the departmental undergraduate admissions effort. Admission to the department is highly competitive. There are three paths to admission into the major. All paths use the same criteria for admission: grades in prior courses, especially in math, science and engineering, a written personal statement, and potential to contribute to the field of computing. In the standard process, students apply to the program after completing a set of program prerequisites. Prerequisite courses include one year of calculus (Math 124, 125, 126 or 134, 135, 136), two quarters of programming (CSE 142 and 143), 10 credits of science (Physics 121 and 122) and 5 credits of English Composition. An Admissions Committee, comprising of four faculty members (including the Undergraduate Program Coordinator), three undergraduate advisors, and a graduate student representative evaluate the applications. Three readers, including two faculty members and one advisor, read each application prior to the committee meeting. The committee makes all final decisions by consensus.

All departmental admissions forms are on-line and students can complete the process remotely. Once the on-line application is completed, students receive email confirmation of their submitted application, and then receive notification from departments about 3 weeks after the application deadline (July 1 for Autumn admission and Feb 1 for Spring admission).

Since 1999, the department has also offered about 40 admission slots to highly competitive entering freshmen that have specified an interest in the department on their application to the UW. These Direct Admission students are coded as majors immediately so that they can take advantage of all department resources from day one.

We recently added a new path into the department called Accelerated Admissions, which allows high-achieving students from CSE143 to apply early to the majors. This was

done to minimize the barrier of entry into our program for particularly strong students, similar to the Direct Admission program for entering freshmen. This has been a very successful program and in attracting students who may not have otherwise completed the requirements for the normal admissions process. It is important to note that students admitted through Direct or Accelerated Admission must still complete the program prerequisites.

Our department is very active in recruiting the best students for our program. We host information sessions every three weeks during the academic year in order to introduce prospective students to our department. We also present at local community colleges, either through in-classroom visits or as part of transfer "fairs," and as needed for various on-campus constituencies.

In 2004 we began offering two seminars geared toward engaging more women in computing careers. One seminar is for students who have no experience with programming. The second seminar is for students to take in conjunction with our introductory programming courses. The goal of these seminars is to help get the message out that a degree in CSE is more than "just programming". Several students from the second seminar became majors and were instrumental in developing a women's chapter of the ACM, called the ACM-W, within the department. ACM-W members serve as mentors for pre-major women and additionally they hold lunches, movie nights, and panels throughout the year to help improve the community of women in our department.

In 2006 we created a team of CSE Ambassadors. Ambassadors are undergraduate and graduate students who are trained to give various outreach presentations to local high school and middle school students. We look forward to seeing this program continue to develop over the next academic year.

Additionally, in 2006 we created a set of videos geared toward improving the image of CSE in the broader community. These videos, which can be found at: <http://www.cs.washington.edu/WhyCSE> have been very well received within the CSE community nationally and internationally. These videos are yet another example of our department's commitment to helping the public learn more about our field. In 2007, one of our videos, "A Day in the Life" was nominated for a Northwest Regional Emmy Award.

New Student Orientation

All new students are required to attend two new student events. The first orientation is run by the advising staff and mainly focuses on familiarizing students with policies and procedures shortly after they have been admitted. It is run by the advising staff and takes place shortly after the student is admitted to the department. There is a second 'welcome event' hosted by our two ACM chapters, where students meet their upperclassman "mentors" who show them the ropes from a student perspective and offer to serve as a resource during their adjustment to the department.

Direct Admission students participate in a one credit seminar fall quarter where they receive more individualized advising and hear from faculty about various research projects taking place within the department. All new students are provided with mentors among our more senior students to help answer their questions and show them the ways of the department.

Student Knowledge of Policies and Procedures

Upon entry to the major, each student is asked to read through the online version of our CSE Student Handbook. This guide details the course requirements for the degree, including prerequisites and curriculum. It also outlines the program's satisfactory progress policy, which specifies criteria for rate and quality of academic progress. The advising staff makes extensive use of email and electronic bulletin boards to remind students of deadlines; to explain curricular, course scheduling, or registration issues; and to provide information about scholarships, internships, and other opportunities that become available.

Advising

The staff maintains an interactive, online database in which transcript information and other data relating to the student's academic progress is maintained. This information is used to record and monitor each student's academic progress and to verify that all program requirements will be satisfied by the expected graduation date. The database includes complete academic records, photographs, and comments entered by the advising staff after every meeting with the student. Faculty are given a separate online feature that shows student pictures, class standing, and links to their class lists. These resources, accessible to our full advising staff, provide a way to track interactions with individual students.

Student evaluation

Students are assigned a decimal grade between 0.0 and 4.0 for each course they complete at UW. Instructors, in cooperation with their teaching assistants, are responsible for grades. There are no formal guidelines for the assignment of grades and most of our faculty do not grade on a curve. Our average course grade for undergraduates is 3.2-3.4 depending on the course. This may seem inflationary but is quite consistent with the high GPAs of our incoming students (approx. 3.5). The minimum grade of 2.0 is required for a CSE course to be used toward the program's graduation requirements. Students not achieving that grade must repeat the course. Our satisfactory progress requirement is that students must successfully complete three courses towards their graduation every quarter (we do not specify a specific number of credits due to the difficulty of collecting a specific number of credits when credits per course vary widely from department to department).

Student mentoring

The undergraduate student organization (the local chapter of the Association for Computing Machinery – ACM) is the principal vehicle for student mentoring. This student-run organization, working under the guidance of the Advising Office, holds semi-annual orientation events that coincide with the fall and spring quarter entry of new majors. This evening event has near universal attendance on the part of new students and the program serves to educate them about how the department is run, what the faculty considers the most important steps and outcomes, and how they can go about obtaining the full variety of educational experiences open to them.

Current undergraduate majors volunteer to mentor new students. They meet at the orientation event and the mentor serves as an informal advisor for the new student's first few quarters in the department. The objective is to help students assimilate the culture of

the department and learn about the resources available to them as well as how to best approach faculty with their concerns.

No formal process is in place for faculty mentoring. However, our students are strongly encouraged, from their first day in the program, to take part in undergraduate research and teaching assistantship opportunities. The number of students involved in our teaching assistant program has increased substantially in recent years. In 2004/2005, 45 undergraduates filled 70 TA positions, in 2005/2006, 96 undergraduates filled 153 positions and in 2006/2007, 117 undergraduates filled 191 positions. This large increase resulted in part from increased undergraduate TA hiring for 142/143 and more grad students receiving funding from other sources. At the same time, complaints from students about TA quality has decreased over the same time frame, largely because of the careful attention given to recruiting and training undergraduate TAs.

In both TA and research assistant positions, students work closely with faculty (and their graduate students) to learn more about other aspects of the department outside of their coursework. The advising staff also works closely with faculty so that they can recommend the right people to contact when students express interest in certain areas. For example, if a student has an interest in computational biology, advisors will often send an email introduction to the faculty and student to help bridge connections. Students often receive advice about graduate school, industry, and research directly from the faculty who teach their courses.

Student Progress Towards Graduation

At the end of each quarter, the advisors review students' grade reports to assure compliance with the satisfactory progress policy. Every student determined to be out of compliance with the satisfactory progress policy is required to meet with an advisor who recommends corrective action. Students out of compliance with the program are placed on probation. Subject to confirmation by the Undergraduate Program Coordinator, students on probation for two consecutive quarters or three cumulative quarters may be transferred out of the program. This occurs on very rare occasions and only after several attempts to remedy the situation.

If students are having academic difficulty in CSE classes, the advisors recommend they seek tutoring assistance from our volunteer tutoring program that is staffed by current graduate and undergraduate students. Tutoring may include individual tutoring by the student volunteers, or help in forming a study group with other CSE majors.

Petitions

Exceptions and substitutions regarding prerequisites or degree requirements are rarely granted, and then, only with the explicit permission of the Undergraduate Program Coordinator. If a student has taken classes at another four-year institution and wishes to transfer credit, they must file a petition.

Graduation Application Process

Students may graduate at the end of any academic quarter. Satisfactory completion of the Computer Engineering curriculum is assured by an auditing process and by

encouragement to regularly meet with the advising staff. A special checkpoint meeting takes place two or three quarters preceding the proposed quarter of graduation. Each student must file a formal degree application with an advisor (who also checks with the University of Washington's DARS - Degree Audit Reporting System – to ensure compliance). This meeting includes a plan of courses in progress and those scheduled for the remaining quarters leading up to graduation. The outcome is that each student is required to have an acceptable terminal course plan over two quarters in advance of their actual graduation.

Acceptance of Transfer Students

Transfer students first apply to and are admitted to the University through the University admissions process, and then separately apply to the Department through the same admissions process as students at UW. Our advising staff makes annual visits to local community colleges to hold information sessions that address the specific issues faced by these students.

Under the state Higher Education Coordinator Board agreement, students may transfer a maximum of 90 credits towards their degree. Students from Washington State community colleges may petition to have 135 transfer credits applied to their degree. In all cases, the final 45 credits of the degree program must be earned while in residence at the University of Washington.

The department also permits the transfer of credit for any course that is part of our major requirements. If the course the student completed at another institution is on our list of pre-approved transfer courses (primarily, from the state's Community Colleges) then credit is granted immediately. If it is not, then the student is asked to go through the petition process described below in the Evaluation of Transfer Credit section. The outcomes can be that transfer credit is granted, denied (and the student must take the course in our department for credit), or granted under the condition that the student makes up for some gaps (this can take many forms including, but not limited to, an informal conversation with the faculty member evaluating the request, direct study to make up for slight differences to our courses, or the completion of previous course exams and/or assignments). This highly individualized approach is only practical because it is rare. The overwhelming majority of our transfer credits are from Washington State Community Colleges with which we have course articulation agreements. Most other transfer credits are from other four-year institutions – usually out-of-state. Courses in engineering that come from an EAC/ABET accredited program are usually accepted readily. Other courses, including courses internationally from other than "substantially equivalent" programs or from CEAB programs in Canada, are considered with great care.

Evaluation of Transfer Credit

Transfer credit is accepted subject to the evaluation of each course. The UW provides a centralized evaluation process that we use for non-computing courses. For each course in the major, a petition is submitted that includes copies of a syllabus, homework assignments, exams and the name of text/s used. There are faculty leads for every 300 level course who review the petition and accompanying documents and make the final determination on whether transfer credit should be granted. The two courses that most often fall into this category are our introductory programming courses. The department has a program for the State's community colleges to have their course pre-approved

(details can be found at:

www.cs.washington.edu/outreach/CC/cc_transfer_application.html). We work closely with community college instructors at annual workshops to ensure their introductory programming courses are well-aligned with ours.

Student Achievement

Our students annually garner a large number of awards and prizes. Over the last 5 years, these have included a number of national and University awards:

National Awards

- Two Rhodes scholar nominees
- Four Barry M. Goldwater Scholarships, one honorable mention
- Two Marshall Scholarship nominees
- One Gates Cambridge Scholarship
- One Google Anita Borg Scholarship
- Two winners, **five** finalists, and **16** honorable mentions for the CRA Outstanding Undergraduate Award (only two winners are awarded each year)

University Awards

- Three Freshman, Sophomore, Junior Medalists, awarded annually to the three highest-performing freshman, sophomore, and junior at the UW.
- Three recipients of the Dean's Medal, which recognizes the top student in each of the College of Arts & Science's four divisions.
- Two recipients of the Dean's Medal, which recognizes the top two students in the College of Engineering.
- One Sterling Munro Public Service Fellowship

A more indirect measure of our success is the admit rate of incoming students, who overwhelmingly "vote with their feet" for our program: More than 95% of the students we accept choose to join our department over other departments, which speaks to our reputation as one of the best programs at the University of Washington. For a complete list of all awards see: <http://www.cs.washington.edu/education/ugradscholars/allawards.htm>.

Evaluation and Improvement

We recognize that the quality of our program depends in large part on the quality of our students, and we expend a great deal of effort trying to attract the best students. We track the quality of our applicant pool and the students we accept to ensure that we maintain a high caliber student-body. The advising staff tracks the statistics for each admissions cycle, listing the GPA of applicants and those we accept and admit.

Our statistics showed a gradual drop in overall interest in the areas of computer science and engineering, starting in about 2001. While we were not yet seeing an effect on the quality of our students, we reacted aggressively in a number of ways to increase the number of applicants and to attract a wider range of students to the program. This included starting two seminars geared toward supporting women in the introductory programming courses, meeting with various units on the UW campus to ensure the proper messages were being sent about our department as a whole, and participating in an increased number of outreach opportunities with local Seattle public schools.

Applications	2004-2005	2005-2006	2006-2007
Total Applications	324	289	322
Applicants Offered Admission	142	148	154
% of Appl Offered	43.8	51.2	47.8
Applicants Who Enrolled	138	141	152
% of Offers Enrolled	97.2	95.3	98.7
Average Cumulative GPA of Applicants	3.45	3.43	3.44
Average Cumulative GPA of Offers	3.66	3.59	3.67
Average Cumulative GPA of Enrolled	3.66	3.59	3.67

In addition, the revamping of the Introductory Programming courses, starting in 2004, has had a beneficial effect on the applicant pool. We are seeing more women applicants as well as students who were not interested in computing until taking the Introductory Programming course series. We have coupled this increased interest at this level with the Accelerated Admissions program which allows students who perform well in CSE 143 to apply early to the department.

Thus, even though the waning of interest in the computing fields has continued, a trend that is even more exaggerated on the national scale, the statistics given above show that the quality of our incoming students has remained consistently high, in spite of a drop in the size of the applicant pool. We hope and expect that this trend is only temporary and that there will be a renewed interest in computing as students understand how central computing will be to every field in the future.

In 2006 our advising unit began maintaining a Wiki resource that is used for documentation of policies and procedures, tracking problems that arise with certain courses, exceptions made for individual students and curriculum changes. The Wiki has been extremely helpful for tracking purposes by making sure we close the loop on any student concerns that are brought to our attention.

2 Program Educational Objectives

Our Program Mission Statement is:

To educate our students so that they will reach their full potential in computer science and engineering research and industrial practice through a deep understanding of the fundamentals of the field, their application in solving problems and creating products, and with an affinity for lifelong educational renewal.

The mission statements of the College of Engineering and the University of Washington are entirely consistent with our departmental educational mission. These are not repeated here for the sake of brevity (the mission of the University of Washington is given in the Faculty Handbook, Vol. 4, Ch. 1, and on the web at: www.washington.edu/faculty/facsenate/handbook/Volume4.html, and the mission of the College of Engineering is available at: www.engr.washington.edu).

Our mission statement leads to the following program objectives in four areas. These are published on our department ABET Web page: www.cs.washington.edu/education/ABET/

Engineering Quality: Our graduates will engage in the productive practice of computer engineering to identify and solve significant problems across a broad range of application areas.

Leadership: Our graduates will engage in successful careers in industry, academia, and public service and attain leadership positions where they have impact on their business, profession and community.

Economic Impact: Our students will enhance the economic well-being of Washington State through a combination of technical expertise, leadership and entrepreneurship.

Lifelong Learning: Our graduates will adapt to new technologies, tools and methodologies to remain at the leading edge of computer engineering practice with the ability to respond to the challenges of a changing environment.

Department Constituencies

Our primary responsibility as part of the University is education and our most important constituents are our students, past, current and future:

- students currently enrolled in our major,
- our alumni,
- students seeking to become majors in our programs.

The organizations that directly rely on the students we graduate and their knowledge, abilities and skills form the next set of constituents:

- prospective employers, including the member companies of our industrial affiliates program. Our Affiliate Program currently comprises more than 50 companies including Amazon, Adobe, Apple, Ford, Google, Expedia, IBM, Microsoft, and Intel. Several national labs (Sandia, Pacific Northwest National Lab, etc.) and several

venture capital firms (Vulcan, Madrona, Granite, etc.) are also affiliates. The current list is online at <http://www.cs.washington.edu/affiliates/curraffil.html>. There is an annual Affiliates meeting where representatives from member companies interact with faculty and students to learn about new research results, forge new research relationships, interview prospective students for internships and permanent positions, and give advice on both research and educational issues.

- university graduate programs, including the nation's leading institutions.

Finally, the largest, and most diffuse, group of constituents is:

- parents of past, present, and future majors,
- professional societies and accreditation organizations
- citizens of the State of Washington, and by extension, the State Legislature and Executive.

Each of these constituencies participates in the process of reviewing our educational objectives and evaluating our success in meeting them. This is done through interaction between the constituencies and the department by means of surveys, interviews, and review sessions, as well as informal interaction such as Open Houses and other interaction with parents and the general public. Interaction between constituencies, for example, between students and employers, is also part of the evaluation process. Students interact directly with affiliates and other employers, and this often brings about an evaluation of our program with feedback to the department both through the students and our affiliates.

Our current students provide constant feedback to the department in a number of forms. Student evaluations, which comprise both numerical scores and written comments and suggestions, are taken very seriously. Faculty are expected to address student concerns in their self-evaluation and these frequently lead to course improvements. We also monitor the performance of students in the more advanced courses to ensure that the prerequisite material has been adequately covered. Where we find problems, we make changes to ensure coordination of these courses. Students also meet with the department chair and the undergraduate program coordinator and advisors for lunch each quarter where students ask questions and raise concerns about the program. Examples of problems that have been raised and addressed this way include infrastructure issues in the labs, the workload in some courses, and the need for classes that cover increasingly important topics like security and HCI. This meeting also allows the chair to inform students of new initiatives and changes and solicit questions and comments. This ensures that students are current on departmental plans and that we know about issues that concern the students.

In addition, the Undergraduate Advising Office and the Undergraduate Program Coordinator are continually in contact with our primary constituencies and forward issues that merit attention to the Curriculum Committee. When deficiencies or other issues are identified the Undergraduate Program Coordinator and the advisors work with the relevant faculty to produce a workable solution and, if needed, bring it before the Curriculum Committee and possibly the entire faculty.

Process used to Review and Revise Educational Objectives

Our program objectives were originally established in 2000 by a process that involved extensive discussions with faculty, students, the advising staff and the College of

Engineering ABET Coordinators Committee. Over the last two years, we conducted a review of our objectives with the goal of aligning them more closely with the ABET definition for objectives. This review process included our industrial affiliates, faculty and students. We held a meeting at our 2006 annual affiliates meeting where we proposed a number of possible objectives and discussed alternatives with the affiliates. We then held a similar discussion in a faculty meeting that resulted in a consensus. A small committee then incorporated these discussions by the affiliates and faculty and formulated a set of program objectives, which were then presented at a faculty meeting and published on our Web for further comment. We then met with current students to discuss the objectives, which resulted in further changes. This resulted in the new set of objectives that are included in this report and published on our Web at www.cs.washington.edu/education/ABET/. We plan to present these objectives for review on an annual basis to faculty, students, affiliates and alumni. This will serve an educational process for those new to our department and allow us to refine the objectives further.

Achieving Educational Objectives

We enable our students to meet the educational objectives by giving them an excellent education in math, science and engineering along with considerable practical design experience. The program outcomes enumerated in the next section are the set of knowledge, skills and abilities that our students achieve in our program and give them the tools to meet the objectives. These are realized through an extensive, carefully coordinated curriculum taught by an outstanding faculty, with provision for considerable practical experience gained through extra-curricular activities.

The curriculum design and its rationale are described in detail in Section B.4 (Professional Component), and details of the courses appear in Appendix I. The curriculum comprises a core set of courses that convey engineering, science, and mathematics knowledge, a second core containing the fundamentals of computer science and engineering, and a required set of advanced courses that apply these fundamentals in a number of areas comprising both software and hardware systems. A capstone design course completes the curriculum with an open-ended design experience that includes team participation as well as oral and written communication of their work. Technical writing courses serve to develop and exercise written and oral communication abilities. Courses in the humanities and social sciences expose students to a breadth of issues including contemporary societal concerns and some of the effects of technology.

In addition, a specific faculty member is responsible for each of the core courses, and is responsible for the syllabus of that course as well as coordinating the professors and instructors that teach the course to ensure consistency across different offerings.

Assessment of Success Achieving Objectives

Listed below are our objectives and the assessment we have defined for each one. The results of our assessment is summarized first, and then the data on which we base our assessment is described in more detail.

Engineering Quality: Our graduates will engage in the productive practice of computer engineering to identify and solve significant problems across a broad range of application areas.

Assessment method: Employer satisfaction with performance, reflected in direct feedback and competition for hiring our students, and admission of graduates into the top graduate schools.

Result: Our students are in high demand by leading technology firms both nationally and locally. Feedback from our Industrial Affiliates on the engineering ability of our students is extremely positive.

Leadership: Our graduates will engage in successful careers in industry, academia, and public service and attain leadership positions where they have impact on their business, profession and community.

Assessment: Survey of positions and activities of graduates. We expect that a number of our graduates will have achieved positions of leadership.

Result: A good proportion, 10%, of the alumni responding to the survey reported being in management or executive positions. Of those who have been in the workforce for more than 6 years, the percent rises to 25%.

Economic Impact: Our students will enhance the economic well-being of Washington State through a combination of technical expertise, leadership and entrepreneurship.

Assessment: Employment in local companies, roles in new companies and companies moving into the area.

Result: 60% of the respondents to the alumni survey are employed in Washington State at a combination of large companies like Microsoft, Adobe, Amazon.com, Boeing, Fluke and Google as well as small local startups like Impinj, Zillow, Isilon, Teranode and F5 Networks. Google recently opened a local research group in part to attract our students.

Lifelong Learning: Our graduates will adapt to new technologies, tools and methodologies to remain at the leading edge of computer engineering practice with the ability to respond to the challenges of a changing environment.

Assessment: Survey of graduates with respect to professional development and level of involvement with professional societies such as attending conferences and workshops, taking courses and learning new skills and methodologies.

Result: Well over half of the alumni surveyed indicated that they were continuing to stay abreast of new technology through mechanisms like attending conferences, learning new languages and methodologies, and participating in in-house training. 70% of alumni reported they were planning on, working on or had achieved an advanced degree.

The real measure of our success is the success of our graduates, and this forms the basis of our assessment method. By this measure, we have been extraordinarily successful. Our graduates are highly sought after by leading industrial firms like Intel, Microsoft, Google, and Amazon.com. We are among the top 5 suppliers of new technical employees to Microsoft, Google, and Amazon.com (along with MIT, Stanford, Berkeley, and Waterloo). We have approximately 60 alumni now at Google. We also are the major supplier of technical employees to a number of high-tech startups in the Seattle area, such as Impinj, Zillow, and Isilon. In fact, our students are inundated by recruiters at information sessions and job fairs organized by our ACM chapter and Affiliates program.

Many of our students are dual majors. In recent years, the University of Washington has amassed an extraordinary record in the international Mathematical Contest in Modeling. This year, 11 3-person teams from a field of 949 entries were chosen as "Outstanding Winners" in the MCM. Two of these teams were from the University of Washington, and each of these teams included a CSE major among its three members. Each year about

15% of our students go on to top graduate schools. For example, 7 undergraduate alumni are currently graduate students at MIT. Some of our students have also gone on to law school and medical school. Many of our students are entrepreneurial. Routinely, winning teams in the University of Washington Business Plan competition include CSE students. Isilon, one of the most successful information technology IPOs in the past few years, was co-founded by our undergraduate alumnus Paul Mikesell.

We survey our graduates immediately as they graduate via exit surveys to determine where they will be employed or in graduate school. We were able to track 58 of our 142 graduates from the 2004-2005 class: 15 are working at Microsoft, 12 at Amazon.com, and 8 at Google. 13 are in graduate school, including MIT, CMU, Columbia, Cornell, Wisconsin, Harvard Law School, UW EE, Colorado, UCLA, and UCSD. We tracked 125 of our 158 graduates from the 2005-2006 class: 15 are working at Microsoft, 11 at Google, 81 at other companies such as Zillow.com, Amazon, Avanade and 18 went on to graduate school.

We make an effort to track our alumni and send them surveys at regular intervals. These surveys are aimed in part at assessing whether they are fulfilling the program objectives. The most recent survey was performed in April 2007. 160 total students responded to this survey, which represented a response rate of just over 25%. The following tables summarize the data from this survey.

Years since graduation	Number	Work Location	Number
0-5	125	Washington State	96
6-10	24	US	53
>10	11	Foreign Country	11

Current Position	Number	Company	Number
Masters Student	7	Microsoft	16
PhD Student	18	Google	17
Other Student	4	Amazon.com	9
Developer/Engineer	110	Intel	4
Engineering Management	15	Other	114
Executive	3		
Currently Unemployed	2		
Other	1		

Professional Development	Number	Advanced education	Number
Attended conferences	67	Currently a student	18
Authored papers	29	M. S. Degree	16
Learned new programming language	96	Ph.D. Degree	20
Other, e.g. in house training	14	Other Degree	13
Member of professional society	44		

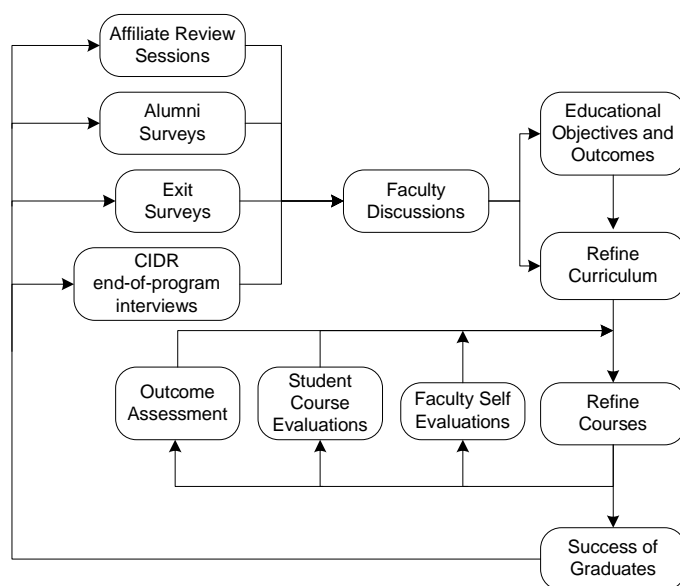
The graduate schools attended included MIT, Carnegie-Mellon, Stanford, UCSD, University of Maryland, and University of North Carolina. In addition, 58 students indicated they planned to return to school for a Masters degree in the future.

On the whole the comments about the department and their experience here were very positive. When asked about what, in retrospect, the department did particularly well students mentioned excellent teaching and facilities, a good balance of theory and practice, a “hardcore” engineering education with many lab experiences, the coordinated curriculum, access to research experience, being taught how to learn, strong capstone courses, and exposure to a broad range of topics.

When asked what the department should improve, students commented that it would have been useful to have larger group projects, more emphasis on internships and research experiences, more practice on presentation skills, more advanced courses, and larger and more complex software engineering projects. Some of the issues raised have been addressed since the students graduated, while others are issues that we are sensitive to and are trying to address within the constraints of the undergraduate experience.

We also receive feedback on our graduates directly through our Industrial Affiliates Program – current and prospective employers are strongly encouraged to join our department’s Industrial Affiliates Program and attend an annual meeting. We have strong ties to these companies (including Microsoft, Intel, Google, Amazon, IBM and others). This key set of stakeholders of major and local companies hire the bulk of our students. At the annual Affiliates meeting, technical representatives of our affiliate companies meet and recruit our students in a setting where students have contact to technical staff rather than just human resource specialists. We explicitly ask our affiliate companies for feedback on our program based on their experience with our students. For example, we have organized a special session as part of our annual Affiliates meeting where representatives from companies meet with faculty and students to discuss the strengths and weaknesses of the students graduating from our program. The feedback we receive about our students is strongly positive, and this is backed up by the competition among these companies to hire our students.

There are many processes in place for evaluating how well we are meeting our objectives and using this evaluation to improve the department. These feedback loops ensure that each part of our program is working the way we have defined, and that the overall mission and goals of the department are being fully met. The overall process is summarized in **Figure 1**.



The large feedback loop shown in this figure evaluates the program as a whole from the standpoint of our objectives and how well our curriculum is preparing students to meet these objectives. The loop uses feedback from graduating students, Alumni and our industrial affiliates. Students are interviewed by staff from the Center for Instruction Research and Development near the end of the program to ascertain to what extent they feel they have successfully met the educational outcomes. This interview format by someone outside the department encourages an open evaluation of the overall curriculum design and the extent to which our educational goals are being met. This is particularly helpful in making students feel that their voice can be heard, taken seriously, and have a positive effect. CIDR produces a formal report to the department each year summarizing this feedback, which is used by the Undergraduate Program Coordinator and the department to address specific issues. Students also participate in an online exit survey which includes both a qualitative and quantitative assessment of our program. This survey provides an independent source of information from our graduating students. Our alumni surveys get feedback from our students after they have had experience in the workplace or graduate school and can reflect better on how well the department has prepared them to meet the educational objectives. Finally, we get feedback from our Affiliates on a regular basis on how well we have prepared our students and where we can make improvements.

This feedback is used to evaluate the effectiveness of the overall curriculum in meeting our objectives and student outcomes. The Undergraduate Program Coordinator continually evaluates the overall program together with the Undergraduate Advising Office and the Curriculum Committee. There is no set timetable for evaluation. However, the staff of the Advising Office meets weekly with the Undergraduate Program Coordinator to discuss issues that have arisen. The Curriculum Committee meets once or twice per quarter to discuss curriculum changes as well as issues forwarded by the Undergraduate Advising Office that are likely to require further attention and further faculty discussion. The faculty is involved in this process through discussions at our annual faculty retreat and faculty meetings. This process has led to many changes in our curriculum and the way we run the program. Examples are presented in detail in Section 3.

The smaller feedback loop is used to evaluate the effectiveness of individual courses in contributing to the overall curriculum objectives and achieving student outcomes. Faculty members perform most of the assessment of student outcomes as described in Section 3. These assessments are coordinated by the advising staff and Undergraduate Program Coordinator and periodically reviewed to determine where improvements need to be made. Many issues can be handled directly by the advising staff. More systemic or problematic issues are forwarded to the Undergraduate Program Coordinator in weekly meetings with the Undergraduate Advising Office staff. Next in line are the department's curriculum committee and executive committee for issues outside the curriculum. Recommendations from these two committees and the Chair may be brought to the entire faculty when further discussion or possibly a vote when required.

The currently enrolled students give us feedback in many ways and at several points in their student career. Part of this feedback is explicitly organized. For example, all students evaluate our teaching and courses every quarter, and this feedback is used by teachers to improve the course material and the way it is delivered, and by the department to identify courses that may require special attention and teachers that can benefit from further teaching development. Instructors also write self-evaluations of their courses which are included in annual peer reviews of teaching. We have found this feedback invaluable for improving teaching and sharing ideas among faculty.

Other feedback is more informal. Students interact with faculty and advising staff through individual meetings and many special events, such as our quarterly undergraduate lunches and other sessions on specific topics such as research/graduate-school/career planning organized by the advising staff. The lunch meetings with the Chair and Undergraduate Program Coordinator provide the opportunity for students to raise educational concerns and for the department to respond in a timely manner. Many students bring complaints and concerns to the undergraduate advising staff, who collate them and bring them to the attention of the department. Most course Web sites, as well as the advising Web site provide an anonymous email form which students can use to give anonymous feedback. Comments, complaints and suggestions reach the Undergraduate Program Coordinator and, through that position, the Curriculum Committee, Executive Committee, and/or Chair as appropriate. Depending on the seriousness of the issue, it may be dealt with by the advising staff, the Undergraduate Program Coordinator, or passed along to the faculty as a whole for discussion and solution. We have provided many detailed examples of these feedback loops in action in Section 3.

Input from our other constituencies reaches the department in various ways. Information about prospective students reaches the faculty primarily through the undergraduate advising staff and their weekly information sessions for prospective majors. Regular meeting and workshops with Community College instructors provide important feedback on our introductory curriculum (which is mirrored at the state's CCs). In addition, we interact with the University's admissions office when we admit freshmen directly to our major. Through the advising staff's information sessions and our undergraduate admissions process, the department can make adjustments to pre-requisites to the major at UW and interact with K-12 schools and community colleges regarding their preparation of students.

A much larger audience reviews the videos we produce to capture the resulting projects from many of our capstone courses. The audience includes the general public (the videos are broadcast on the UW cable television channel) and prospective students. Comments from all these groups make their way back to the Undergraduate Program Coordinator and the Advising Office and are incorporated with other feedback.

Finally, the citizens of our state have ample opportunities to influence the department through our various outreach and recruitment efforts and their interaction at annual open house events, state legislative representatives, and various community service organizations on which our faculty serve.

3 Program Outcomes and Assessment

Our program outcomes comprise the set of skills and abilities that our curriculum is intended to give our students so they can meet the educational objectives we have set for them. These outcomes are closely related to the ABET outcomes, modified in part to meet our program needs.

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice.
- (l) knowledge of probability and statistics
- (m) knowledge of discrete mathematics

Achieving Outcomes

The computer engineering curriculum has been specifically designed to ensure that the program outcomes are met by every computer engineering student. This curriculum comprises a set of general requirements, core courses in computer science and engineering, outer core courses specific to the software or hardware specialization, and capstone courses. In addition, internships and coops provide real-world experience, research projects allow students to get experience working on cutting-edge research with faculty, and teaching assistantships allow students to experience the teaching side of the profession.

Table 1 summarizes how each element of the curriculum and extra-curricular activities contribute to the achievement of the outcomes. Our curriculum is designed to provide students with a solid foundation and extensive practice in one of two areas of specialization. Table 2 summarizes the contribution of each of the courses in our curriculum to the outcomes. Each course contributes to several outcomes, and outcomes are supported by several courses. An entry of H indicates a high contribution for that outcome and M indicates a medium level of contribution. Table 2 also shows how outcomes are assessed: the shaded entries of the table indicate the outcomes that are assessed in each course using our targeted assessment process. The curriculum is described in detail in Section B.4, the course syllabi are given in Appendix I, and the details of outcome assessment are described later in this section.

Table 1 – Program outcomes and mapping to curricular and extra-curricular elements.

	Curriculum Component (credits)									Extra-curricular		
	General education (42)	VLPA/IS (30)	Communication (12)	Math/Statistics (19-22)	Science (20)	CSE core (47)	HW/SW outer core (22-24)	Capstone	Electives	Industry internships	Research project	Teaching assistantships
a an ability to apply knowledge of mathematics, science, and engineering				M	M	M	H	H				
b an ability to design and conduct experiments, as well as to analyze and interpret data					M	M	H	H			M	
c an ability to design a system, component, or process to meet desired needs within realistic constraints						M	M	H		H	M	
d an ability to function on multi-disciplinary teams							L	H		H	M	
e an ability to identify, formulate, and solve engineering problems						M	H	H	M	M	H	
f an understanding of professional and ethical responsibility						M	L	M		M		
g an ability to communicate effectively			H			M	M	H		M	M	M
h the broad education necessary to understand the impact of engineering solutions	H	H							M	M		
i a recognition of the need for, and an ability to engage in life-long learning							M	H		H	H	
j knowledge of contemporary issues		M				M	M	M		H	H	
k an ability to use the techniques, skills, and modern engineering tools						H	H	H		H	H	
l knowledge of probability and statistics				H								
m knowledge of discrete mathematics						H						

Table 2 – Program outcomes and mapping to curriculum requirements.

		a	b	c	d	e	f	g	h	i	j	k	l	m
Core Courses														
	Mathematics	M											H	
	Science	M	M											
	General Education/VLPA-IS								H		M			
	Technical Communication							H						
CSE 142	Computer Programming I					M						H		
CSE 143	Computer Programming II					M						H		
CSE 303	Concepts & Tools for SW Dev.						H	M	H		H	H		
CSE 321	Discrete Structures	M											H	H
CSE 322	Intro. to Formal Models	M		M		M								H
CSE 326	Data Structures	M				M						M		H
CSE 341	Programming Languages					M						H		
CSE 370	Intro. to Digital Design	H		M		M						M		
CSE 378	Machine Org. & Assembly Lang.	M	H	H		M						M		
CSE 451	Operating Systems	H	H	H	H	H						H		
CSE 461	Networking	H	H	H	H	H						H		
EE 215	Fund. of Elect. Engineering	M	M	L		M						M		
Hardware Option														
EE 233	Circuit Theory	M	H	M		M						M		
CSE 466	Embedded Systems	H	H	H		H						H		
CSE 467	Advanced Digital Design	H	H	H		H						H		
Capstone Design		H	H	H	H	H	M	H	M	H	M	H		
Software Option														
CSE 403	Software Engineering	H	H	H	M	H					M	H		
Two of the following senior courses:														
CSE 401	Intro. to Compilers	H		M		H						H		
CSE 421	Intro. to Algorithms	H				M								H
CSE 444	Databases	H		H		H					M	H		
CSE 466	Embedded Systems	H	H	H		H						H		
CSE 471	Computer Design	H	H	H		H						H		
Capstone Design		H	H	H	H	H	M	H	M	H	M	H		
Senior Electives – Also any courses listed above														
CSE 431	Intro. to Theory of Computation													
CSE 457	Computer Graphics													
CSE 458	Computer Animation													
CSE 468	Introduction to VLSI													
CSE 473	Intro. to Artificial Intelligence													
CSE 498/499	Senior Project/Research													

Our curriculum contains a strong focus on engineering design, starting with small design problems in the introductory programming courses, building on more substantial design problems at the 300-level, course design projects in many of the 400-level courses and culminating with a capstone design experience at the senior level.

Cooperative education, internships, teaching assistantships, and research involvement on campus are ways in which students can further enrich their education with real-world experiences that will

provide a more informed outlook on their future career. We strongly encourage our students to obtain these experiences and we have a very high participation rate.

Of the 157 students in the 2005/2006 graduating class, 33 (21%) completed either college honors or department honors, 118 (75%) participated in a tech-related internship or co-op experience, and 39 (25%) served as teaching assistants. These numbers are representative of participation over the past several years. Finally, for the 2006/2007 class, a total of 89 (~56%) students engaged in research projects with faculty. We feel that this extraordinary level of participation gives almost all our students substantial practice and experience outside the classroom with almost all of the program outcomes.

Summary of Student Achievement of Outcomes

Table 3 summarizes the results of our assessment of how well our students are meeting each of the individual outcomes. We present the details of our assessment process and the assessment data following this summary evaluation.

Table 3 – Summary of Assessment Results for Program Outcomes

Program Outcome	Self Ranking	Evaluation
(a) an ability to apply knowledge of mathematics, science, and engineering	Excellent	Our students perform very well in math and science classes and come to our program with excellent background. Many of our courses require that students apply these concepts in combination with engineering skills to solve problems and complete projects.
(b) an ability to design and conduct experiments, as well as to analyze and interpret data	Excellent	Testing and debugging of computer systems requires substantial simulation and experimentation to measure performance and understand the effect of design choices. Our students conduct such experiments in a range of systems and design classes.
(c) an ability to design a system, component, or process to meet desired needs within realistic constraints	Good	Students gain extensive experience in the design of systems and components in many courses, especially the project and capstone courses. In many cases, these involve solving many different kinds of constraints.
(d) an ability to function on multi-disciplinary team	Good	All students work on a multi-person team in the completion of the capstone design project. In addition, many students complete senior, honors and research projects, where they work in teams. We are working to include better instruction on effective team management, and to better assess individual students' ability to effectively work on a team.
(e) an ability to identify, formulate, and solve engineering problems	Excellent	Our students demonstrate a strong ability to identify and solve engineering problem in a range of courses. Course projects, capstone design projects and research require substantial design and evaluation. Feedback from our students, affiliate companies and graduate schools indicates that our students excel on this outcome.

(f) an understanding of professional and ethical responsibility	Good	CSE 303 (Concepts and Tools for Software Development) focuses in part on a variety of ethical issues and the societal impact of technology. Other courses also present these issues where appropriate. We are working with others in the College of Engineering to strengthen this part of our program by introducing modules on ethical principles and their application in the engineering context.
(g) an ability to communicate effectively	Good	Students take courses in Technical Communications that cover both writing and speaking. A combination of course grades and an assessment of writing samples from CompE courses are used to assess this outcome. In addition, students are required to successfully complete a presentation and a written report in the capstone design class. We are working to find ways to successfully integrate writing into our more of our courses.
(h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context	Good	Our students are required to take a range of courses to meet the VLPA requirement and we rely on these courses to provide a broad education. This outcome is also covered and assessed in CSE 303 and in the Capstone courses.
(i) a recognition of the need for, and an ability to engage in life-long learning	Good	Many of the more senior courses, particularly the capstone courses and senior projects, require students to discover and learn information on their own. In addition, many students attend department seminars and colloquium and take part in research projects with faculty and graduate students. Student surveys confirm that students recognize this ability is important.
(j) knowledge of contemporary issues	Good	Most courses present concepts in the context of the current state of the art. The societal context and impact of work in computer engineering is explored in capstone courses. We are working to formalize the assessment of students' understanding of societal factors.
(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	Excellent	Students become skilled with many different "industrial-strength" tools and use them to complete class assignments and projects.
(l) knowledge of probability and statistics	Good	Statistics 390 is a required course for all students.
(m) knowledge of discrete mathematics	Excellent	CSE 321 (Discrete Mathematics) is a required core course for all students.

Assessment of Outcomes

Many different means are used to assess whether our students meet our program outcomes. The results of these assessments are evaluated by the advisors, the undergraduate program coordinator and other faculty and are used to make changes to the program. Note that the

combination of two programs under one roof, computer science and computer engineering, means that most of our assessment measures are applied to both student populations.

Assessment Tools

The department uses several forms of assessment. Those listed first give us the most detailed data on outcome achievement while those listed later yield a more general overview of overall student performance.

- Targeted assessment of outcomes in courses
- Writing assessment through the Department of Technical Communications
- End-of-program group interviews conducted by the Center for Instructional Research and Development
- Student surveys upon exit from the department
- Surveys related to industry employment (pre/post and employer) through the College of Engineering's Coop Office
- Alumni surveys
- Course grades
- Student course evaluations
- Faculty self-evaluations

The combination of all of these metrics provides the leadership of the department and the faculty with a multi-faceted picture of the ability of our students to achieve the program outcomes.

Targeted Assessment of Outcomes in Courses: Over the past year we have adopted a new method for assessing outcomes in courses. This assessment process replaced our previous procedure that collected samples of student work in courses. Until recently, we relied on reviews of these course portfolios to judge student performance on outcomes. However, we found that this did not target the outcomes specifically enough, and thus we have adopted the new procedure which we believe will provide better information, with less overhead for the instructor.

Several relevant required courses are chosen for the assessment of each outcome so that each course assesses a small number of outcomes, except for the capstone courses, which assess most of the outcomes. The highlighted entries in the course matrix of Table 2 show which outcomes are assessed in the different courses. The instructor of each course decides and documents how the outcomes for that class are to be assessed. Depending on the outcome, assessment may be via a targeted assignment, exam questions, an assigned paper or project, or an assessment of oral or written presentation using a standard rubric. The instructor records the result of the assessment for each student by categorizing the student in one of four categories:

- Exemplary: Performing beyond expectations
- Competent: Performing at expected level
- Developing: Mostly performing at level but need more practice
- Novice: Not performing at the required level

The instructor also selects representative work from the top, middle and bottom of the class. This material, including the assessment description, the assignment or exam used for assessment if appropriate, a class histogram and samples of student work are collected in one place. This

additional information allows assessments by different instructors to be evaluated and compared. This new assessment process has been set up so that each outcome is assessed several times per year across several courses. Appendix 1.D contains a detailed description of this targeted assessment that the faculty use as a reference guide.

The data collected in this assessment is included later in this section in the discussion of each outcome. Since we have only begun this new system, we do not have much data yet. However, as we continue to collect this data, we will be able to use it to analyze trends in our students attainment of outcomes much better than we do at present.

Assessment of writing through the Department of Technical Communications: Our students encounter writing assignments in a variety of courses, both inside and outside the department. The assessment of the writing outcome is currently provided primarily by the required Technical Communications course, by writing in the capstone courses, and by an assessment of writing samples by a departmental committee. These provide direct evaluation of student writing proficiency.

Student course evaluations: Every course in the department is formally evaluated every quarter by the students. Besides the standard evaluation questions, students are also asked to rate how well the course addresses the ABET criteria (A-K). This permits us to verify our mappings of outcomes to courses and ABET criteria. In addition, it will permit us to track how our courses are evolving relative to the criteria over time. Students were asked to what extent the course contributed to their development of the different outcomes, with responses ranging from 0 (not at all) to 6 (very much). The results are tabulated in Figure 2. It is clear from this data that students feel that the outcomes that are not being addressed include (d) being able to work in teams, (f) understanding professional and ethical responsibility, and (h) a broad education. This is understandable since many of our courses do not focus on working in teams, ethics or serve to provide a broad education. For the remaining outcomes, especially the application of math, science and engineering concepts, problem solving and using modern tools, students feel that our courses are doing a good job.

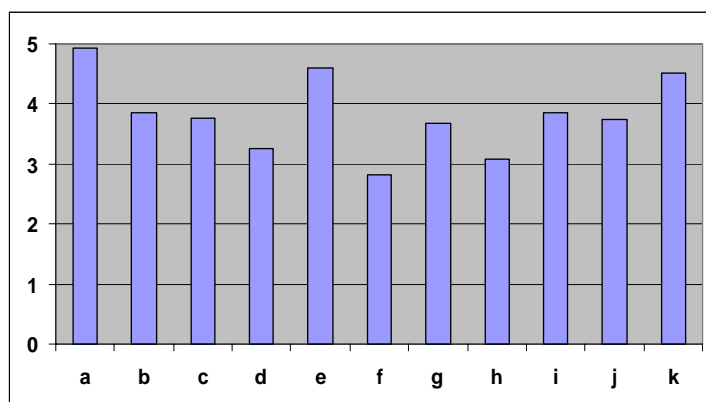


Figure 2 - Student responses to ABET outcomes on course evaluations, 2006/2007

CIDR end-of-program interviews: Starting three years ago, we have been inviting staff from CIDR (Center for Instructional Development and Research) into all our capstone courses each year to hold town-meeting style discussions of our program with the students. This discussion focuses on an

evaluation of our program educational outcomes and solicits specific comments and suggestions for improvement from the students. These are then collated, anonymized by CIDR and reported to the Undergraduate Program Coordinator. The results are reported back to the department chair, executive committee and curriculum committee for remedial action. We have found these interviews valuable for uncovering issues that we have been able to address quickly before they became any larger. Two recent examples of direct action taken in response to these sessions were:

- 2005: There were many negative comments about the quality of TAing. In response, we instituted new procedures for selecting and training undergraduate teaching assistants. This process has worked so well that in the 2006 and 2007 reviews, there were almost no negative comments on TA quality.
- 2006: An issue that continues to receive many student comments is the high workload to credit ratio in many of our courses. In response to this feedback, we performed a study across all courses and instructors to compare the work hours reported by students. This allowed us to identify the courses with the highest workload and take steps with the instructors to rectify this issue. We will continue to monitor this issue both via workload measurement and through student feedback.

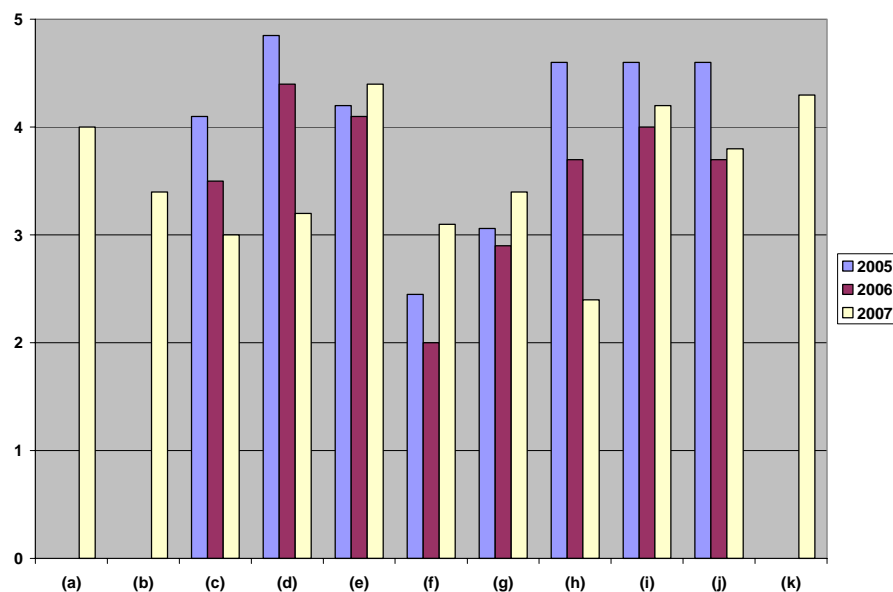


Figure 3 - Assessment data from student interviews in senior capstone courses

The interview also includes a questionnaire that is completed in small groups. The CIDR representative asks the students to discuss and indicate on a scale of 0 – 5 how well they think the department has helped them achieve each of the program outcomes a-k. The student responses over the last 3 years are shown in the graph below. (The questionnaire for 2005 and 2006 did not ask specifically about our current ABET outcomes, but those questions have been mapped to the outcomes in the data below. This mapping was close for outcomes e, f, g, and j, but some outcomes were not covered (a, b, k) and others were less direct.)

The students feel that they are strong on outcomes a, e, i and k, which involve problem solving and design, and life-long learning. We are glad to see that there has been improvement for outcomes f and g, where we have devoted some effort to improve our program. This indicates that our new course, CSE 303, which deals in part with professional responsibility and ethics, has been successful, and that our increased focus on presentation skills is paying off. We were surprised on the most recent survey by the response to outcome (d), working in multi-disciplinary teams and outcome (h), broad education. We will be following up on this to determine why there has been a drop. We believe that it might be an issue with the terminology used in the outcomes.

Student surveys upon exit from the department: Students are asked to complete a Web-based survey which covers many facets of their experience in the department and the University, and asks about their plans after graduation. These comments are collated by the lead advisor and reported to the Undergraduate Program Coordinator and the department chair. Comments on the department are almost uniformly positive. Students are happiest with their senior level courses and capstones, and less satisfied with courses taken outside the department.

College of Engineering's Coop Office surveys: Surveys are given to students before they leave their Coop and after they return. The questions on this survey are listed here and in the information provided on our coop office in Appendix II where the relation of the questions to ABET a-k criteria is also discussed.

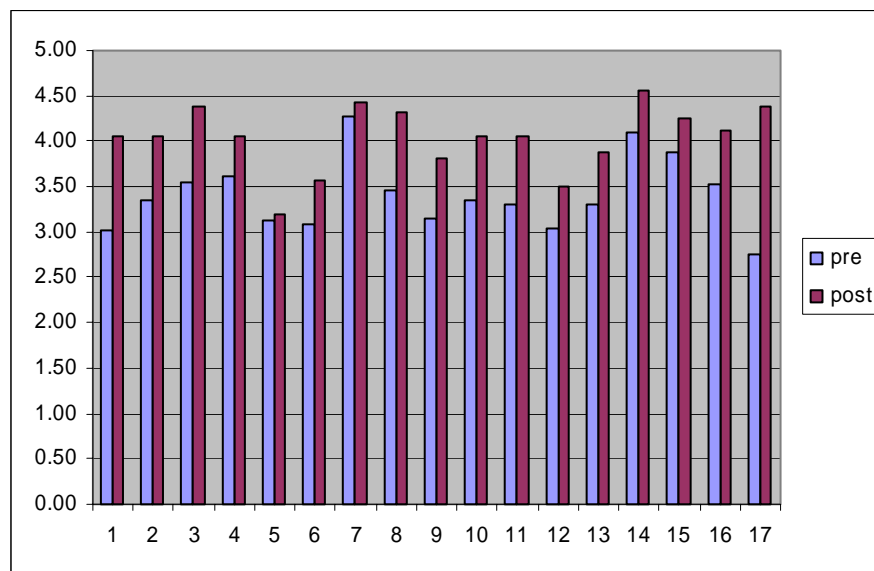


Figure 4 - Coop experience survey 2005/2006

1. Ability to apply engineering theory and principles to the work environment.
2. Ability to design a system, component, or process to meet desired needs.
3. An understanding of professional and ethical responsibility.
4. Ability to communicate effectively on an interpersonal level.
5. Ability to communicate effectively in formal presentations.
6. Ability to communicate effectively in technical writing.
7. Recognize the need for lifelong learning.
8. Ability to use techniques, skills and modern engineering tools necessary for engineering practice.

9. Ability to design and conduct experiments.
10. Ability to analyze and interpret data.
11. Ability to function on multi-disciplinary teams.
12. Knowledge of contemporary issues in your discipline.
13. Ability to identify, formulate and solve engineering problems.
14. Recognize the value of hands-on experience.
15. Ability to think critically.
16. Ability to manage change.
17. Understanding of office dynamics.

We have included the data for the responses for the 2003-2004 and 2005-2006 academic years in the histograms below. It is clear from this data that our students benefit from their Coop experience. The primary areas where students see improvement are on questions 1, 2, 3, 8, 9, 10, 13, 16 and 17. This data shows how students benefit from the coop experience where they are exposed to issues that do not arise in the University setting. As a result, we encourage all students to do an internship or coop, and 80% of our students take this advice by the time they graduate.

Alumni surveys: We survey our alumni to get information that we use in part to assess how well they are meeting our objectives and in part to get their feedback on our program from the vantage point of several years in industry or graduate school. These results were presented in Section 2.

Course grades: Course grades are based on the evaluation and grading of assignments and exams and are an important means for assessing student performance and providing feedback to students and the department. Grades, in combination with direct assessment of outcomes by instructors, provide a direct way for us to evaluate student achievement of many outcomes. The average student grade is approximately 3.4 (out of 4). Although this may seem relatively high, one must bear in mind that the average GPA of our incoming majors is 3.67.

Capstone design projects: The capstone courses are the centerpiece of our program and key to providing and assessing program outcomes. In the capstone course, students work in teams to design and implement a substantial project comprising multiple components. As part of their project, they produce project reports and make class presentations where they present their design, describe the design choices they made, and the tradeoffs they considered. At the end of the course, the project teams demonstrate their projects to their peers, and in many cases to members of affiliate companies and to the department as a whole.

Many outcomes are integral to the student's success in the capstone course, from the mastery and application of fundamental concepts in mathematics, science and computer engineering, to evaluating design tradeoffs and making effective design decisions, to working effectively as part of a team, to self-learning of new concepts and tools, to understanding the impact of computer engineering on society, to effective oral and written communication. The project reports, demonstrations and presentations provide the instructor a direct mechanism to assess whether students have met outcomes b through j. In addition, we often invite visitors from our affiliate and other companies to capstone course presentations and demonstrations, and ask them to help us verify that the projects are at the appropriate level and targeting relevant real-world problems. We also capture many projects from our capstone projects in video productions that highlight the projects as well as the design process and tools the students used in the class. This allows our outside constituencies, particularly our industrial affiliates, to assess the quality and scale of these capstone projects. These videos are available online at www.cs.washington.edu/info/videos/.

Assessment of Outcomes

This section describes in some detail our process for defining, implementing, assessing, and evaluating each of the program outcomes. Performance criteria are defined for each outcome along with the mechanisms for assessing these criteria. This assessment enables us to evaluate whether students achieve the outcomes we have defined, and provide feedback both to the students and to the department.

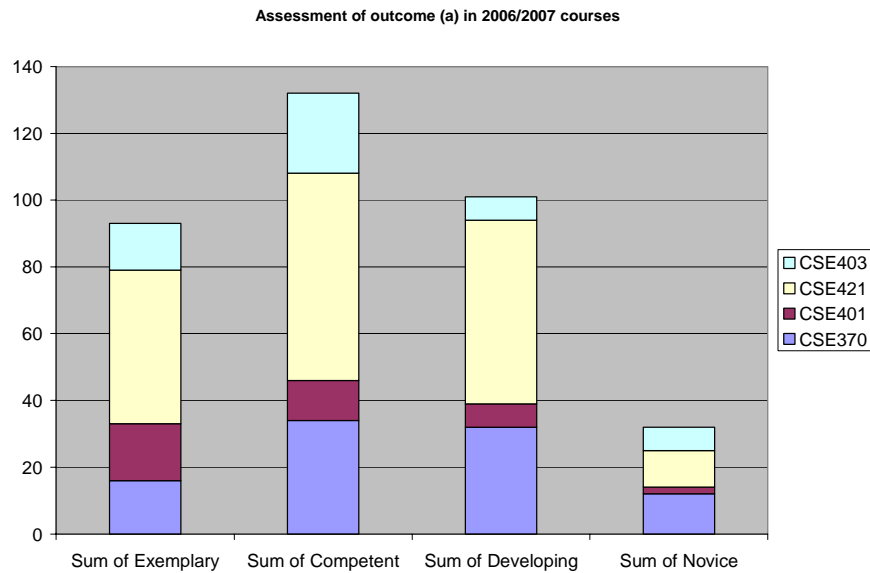
a) An ability to apply knowledge of mathematics, science, and engineering.

Our students enter the program with a very strong background in Math and Science as measured by their performance in required Math and Science courses. For the calculus sequence, our students' average grade for Math 124 is 3.7 (with 1.5% below 2.5), for Math 125 is 3.5 (with 4% below 2.5), and for Math 126 is 3.6 (with 2.3% below 2.5). The average grade for Math 308 (Linear Algebra) is 3.5 (with 3.6% below a 2.5), and the average grade for Stat 390 (Probability and Statistics) is 3.3 (with 6.5% below 3.5). In terms of science classes, the average grade for the Physics 121, 122, 123 sequence is 3.5 (with 2.2% getting below 2.5).

Many of our courses rely on the application of concepts from math and science to solve problems in computer engineering. Mathematics and science concepts appear throughout the curriculum and success in these courses requires students to apply these concepts to solve problems. For example:

- CSE 321 (Discrete Structures) - Students apply many concepts from linear algebra (Math 308).
- CSE 322 (Intro. to Formal Methods) - Students construct mathematical proofs as part of problem solving.
- CSE 341 (Programming Languages) – Proof concepts from mathematics are needed to reason formally about the semantics of programming.
- CSE 370 (Introduction to Digital Design) - Students apply Boolean algebra to solve a wide range of problems. Physics concepts from electricity and magnetism are directly applied.
- EE 215 (Fundamentals of EE) - Many concepts from the calculus sequence, including differentiation and integration are applied to solving circuit problems. Physics concepts from electricity and magnetism are of great importance this course.

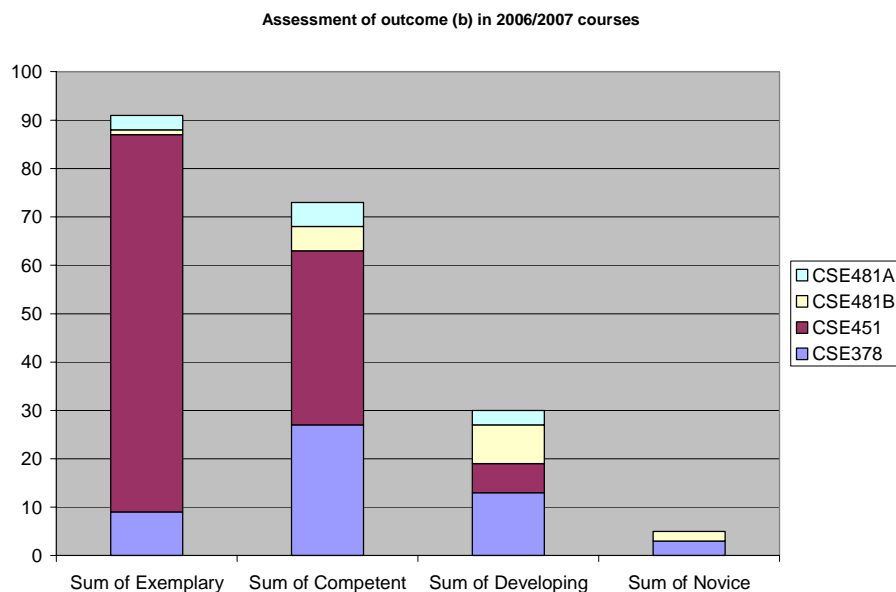
Finally, many upper-level courses require students to apply concepts from math and science. CSE 421 (Algorithms), CSE 431 (Theory of Computation), CSE 457 (Graphics), CSE 461 (Networks), CSE 467 (Advanced Digital Design), and CSE 468 (Intro VLSI) in particular apply a wide range of math and science concepts.



We formally assess this outcome in the following courses: CSE 370, CSE 466, CSE 467, CSE 401, CSE 403, CSE 421, CSE 444, and CSE 471. These have been chosen to ensure that students are assessed in at least three required courses. The result of the assessments performed thus far is shown above. Although the data is limited since we only recently started performing targeted assessments, it confirms our evaluation that our students meet this outcome. Feedback from our alumni and the companies they work for confirm that our students have an excellent background in math, science and engineering and are able to apply these concepts in their work. Student feedback on the ABET outcomes on course evaluations, and responses to the Co-op surveys confirm this assessment.

b) An ability to design and conduct experiments, as well as to analyze and interpret data.

Expertise in performing experimental work is central to many areas of computer engineering. Understanding how a computing system operates under realistic loads requires good experimental design and data measurement. The use of benchmarking as part of this process is an important



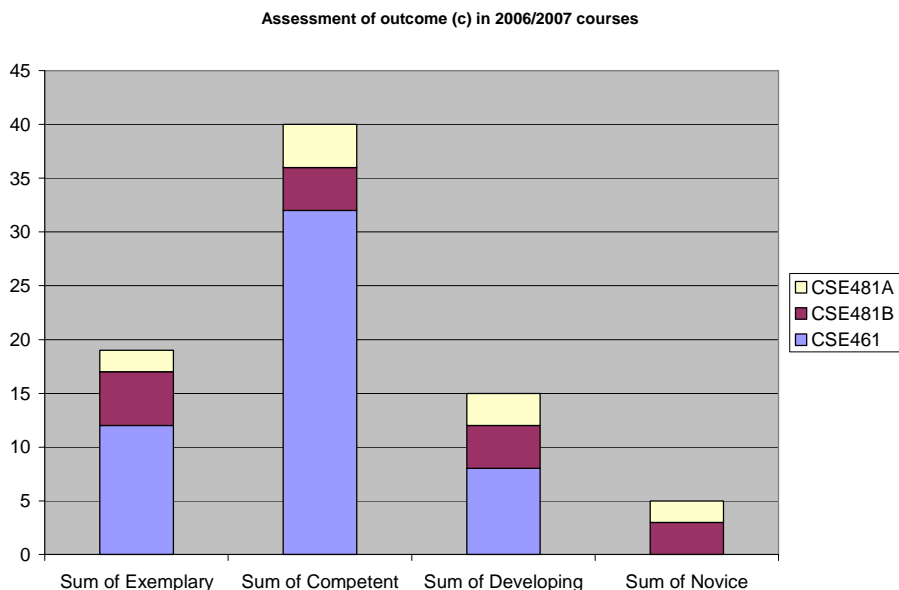
technique, and a quantitative approach to design can be achieved via experiments done using good benchmarks. Simulation is also a very important technique in computer engineering that allows the designer to run experiments on a system before building it. Finally, system debugging can be seen in an experimental framework. Finding problems in a system design requires the careful design of experimental conditions that will expose the problem. Students learn these experimental skills in a range of courses in the curriculum.

We have chosen to assess this outcome in two courses: CSE 378 (Machine Organization) and CSE 451 (Operating Systems), as well as in the Capstone courses. In CSE 378, students perform experiments to determine the effect of different architectural design decisions. For example, the size and organization of a cache can have a large effect on a system, and measuring this effect requires experiments that make use of representative benchmarks. In CSE 451, students perform experiments to understand the performance tradeoffs that are crucial in operating system design. The results of our initial assessments are shown in the graph on the previous page. Note that the senior level students show an increased level of competency of this outcome.

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

Design is central to many courses in our department, as described in detail in Section 4. Students learn to design both hardware and software systems in many course projects designed in senior-level classes. This design experience culminates in the capstone design project where successful completion requires students to demonstrate this ability.

The ability to use abstraction and hierarchy to design complex systems is a topic that appears in multiple contexts in many courses in the curriculum. For example, constructing software solutions to problems in courses like CSE 451 (Operating Systems) and CSE 461 (Networking) requires the ability to break a system into abstract components in the form of methods, data structures and objects.



Constructing hardware solutions to problems in courses like CSE 370 (Intro. Digital Design) and CSE 378 (Machine Organization) also requires the ability to define a hierarchy of abstract hardware components with well-defined interfaces.

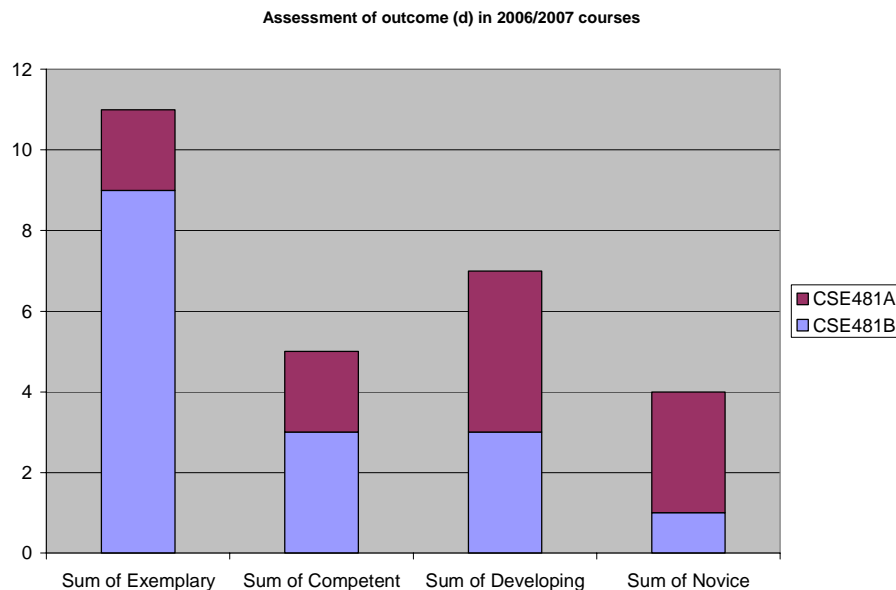
In the capstone courses, students address in their project reports the specific constraints that they needed to meet in the design of their project. We are in the process of proposing a change to the capstone courses to include certain standards for the written reports. We are proposing that all capstone project reports include a section that describes the different alternatives that were considered and the engineering tradeoffs were considered and evaluated.

We have targeted the assessment of this outcome in CSE 378, CSE 461 and in the capstone courses. Students in both these courses complete projects that require constructing a substantial hardware (378) or software (461) artifact and must demonstrate this outcome for successful completion. The results of our first targeted assessment of this outcome is shown in the graph on the previous page. Feedback from students in course evaluation, senior interviews, and exit surveys confirm that they feel technically very well-prepared. Feedback from our alumni and affiliates also indicates that our students are particularly strong in this outcome.

d) An ability to function on multi-disciplinary teams.

Students work in teams during project courses in senior classes, where they learn how to work together with others that have different background, strength and skills. Students demonstrate mastery of this outcome working on a large design team comprising 3-5 students in a capstone course. A component of the capstone courses is coverage of topics on how to effectively work as part of a team and concepts in project planning and management. Project meetings with the instructor and TAs allow the effectiveness of the team to be assessed, and permit problems that surface to be addressed.

This outcome is assessed in two senior project courses, CSE 451 (Operating Systems) and CSE 461 (Networking), in addition to the capstone courses. In the capstone courses, students often work in

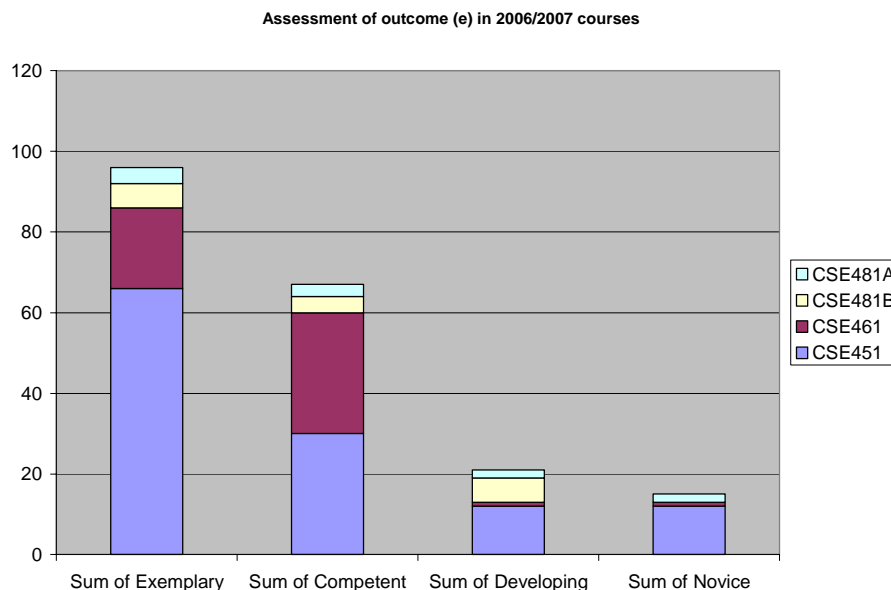


multi-disciplinary teams. For example, in the hardware design capstone (CSE 477), software-oriented students collaborate with hardware-oriented students. In the Urban Sim capstone, our students collaborate with students in the Urban Planning department. Although all students do participate on a multi-person design team, it is difficult to assess the effectiveness and contribution of each student. This is done subjectively by observing team behavior and using an assessment rubric to assess the contribution of each team member. We also objectively assess in terms of the degree of success of the project team. The initial results of our assessment are shown in the graph above. This shows that our students achieve this outcome to some degree, but that there is room for improvement.

Feedback from both our alumni and our affiliate companies confirm that our students could be better prepared to work in teams, which is almost universally required by industry. Of course, in an academic setting, teamwork is often problematic because of the need to assess the contribution of each individual. We are addressing this problem by spending more time on this topic in our project courses, and particularly in our capstone courses. We are first trying to make students aware of how important it is for them to work well in teams. We then discuss topics on team organization and dynamics as well as project planning. We are also continuing our discussion with our affiliate companies on this issue to identify ways to improve this part of our program.

e) An ability to identify, formulate, and solve engineering problems.

Most of our courses, particularly the senior level courses, are all about the application of the concepts of computer engineering to the design and evaluation of solutions to engineering problems. Our department has a very strong belief in learning by doing, and that skill in design is learned best by doing design. Most senior courses incorporate a design project where students directly apply the concepts they are learning to the design and evaluation of solutions to problems. Finally, the capstone course, which is focused on the design of a large, multi-person project, requires the application of concepts acquired in multiple courses for successful completion. Students also demonstrate achievement of this outcome in senior and research



projects, and in co-op and internships.

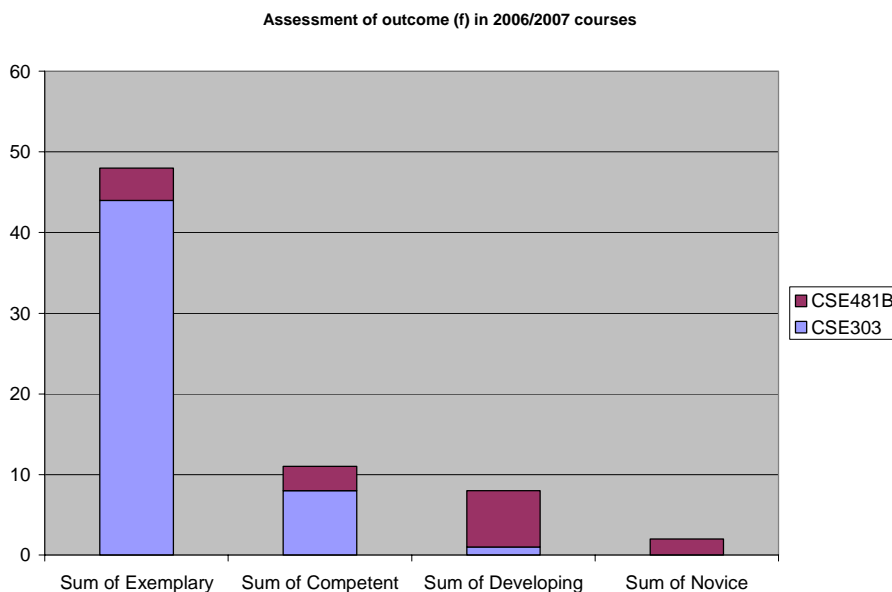
Targeted assessment of this outcome is performed in two senior project courses, CSE 451 (Operating Systems) and CSE 461 (Networking), as well as the capstone courses. The results of our first assessments are shown above. Feedback from alumni and affiliate companies confirm our assessment that our students are very strong in achieving this outcome.

f) An understanding of professional and ethical responsibility.

CSE 303 (Software Tools) has been designed to include a discussion of professional ethics and responsibility. This course also exposes students to a number of areas where computer systems interact with society and raise ethical issues, including privacy and security, software reliability, intellectual property, and electronic voting. Current events are used to motivate these ideas and case studies for class discussion and student assignments. This material is covered via presentations and discussions of specific topics, using current events for motivation and case studies. Students in CSE 303 complete writing assignments where they address ethical issues that they may face in the practice of computer engineering. Students are asked to discuss and evaluate specific issues in these assignments. The instructor assesses the student's understanding based on these assignments.

The capstone courses revisit this topic in the context of the capstone topic area. Many other courses also include coverage of the impact of computer engineering on society, but in a less systematic and formal way. For example, in CSE 451 (Operating Systems) and CSE 461 (Networks), the topics of security and privacy are covered, along with the implications on society of decisions made by computer engineers. In addition, two courses in the Information School, INFO 444 (Value Sensitive Design) and 454 (Information Policy: Domestic and Global), were added as senior electives to add more opportunities for our students to explore implications of computer engineering on society.

This outcome is assessed in CSE 303 and in the capstone courses. We show below the initial results of this targeted assessment. These results show that our students are meeting this outcome. This is



confirmed by student self-assessment in the capstone interviews.

Although the steps we have taken to improve our students' awareness of professional ethics and responsibilities have been effective, we recognize that we need to make further improvements. We are working with the college and other departments to respond this challenge. This will rely, in part, on specific modules used in courses besides CSE 303, as well as courses and seminars developed by the college.

g) An ability to communicate effectively.

Effective communication includes both written and oral communication. Formal writing is addressed and directly assessed in the writing courses offered by Technical Communication, TC 231(Introduction to Technical Writing) and TC 333 (Advanced Technical Writing and Oral Presentation). TC 231 teaches students the fundamentals of writing for a technical audience in the form of reports and scientific papers. TC 333 emphasizes both technical writing and oral presentations. Students are required to make presentations as part of this class. Grades in these courses are consistently above 3.0 with less than 10% below this level. For the two technical writing courses combined, the average grade for our students is about 3.5.

Many of our own courses now include writing, and our students have more opportunities to practice their writing skills in project assignments that require written reports and/or documentation.

Our primary assessment method for writing uses a committee led by a consultant from the Department of Technical Communications to evaluate samples of student writing. This was last performed in the Summer 2006 using samples of student writing from four junior and senior-level courses. The committee comprised of two faculty (Carl Ebeling and Alan Borning) and a senior graduate student (Andrew Petersen) together with Karen Kasonic from Technical Communications. 51 papers were evaluated, 39 from senior-level courses, using a standard rubric and assigned to one of several categories. Table 4 shows the results of this assessment, along with the results from our 2001 assessment. The results are quite similar, except that many more papers were rated as Strong in 2006, and more papers (40%) were rated below Competent. We had hoped to see more improvement given the greater emphasis on writing in our program. Our goal now is to reduce the number of low-ranked papers to at most 30%.

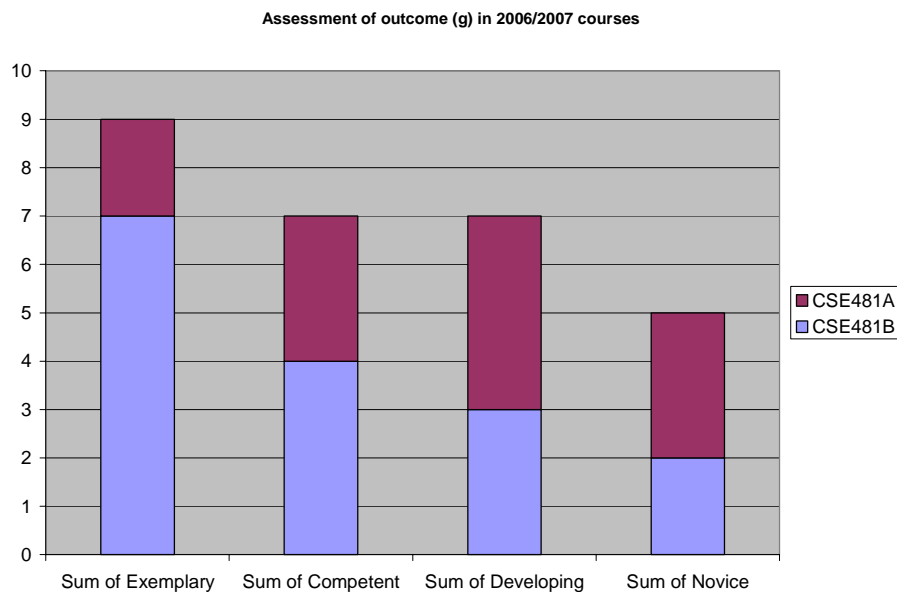
Table 4 - Assessment results for writing

Category	2001 Assessment	2006 Assessment
Unacceptable	13%	10%
Weak	N/A	18%
Low Competent	9%	12%
Competent	53%	35%
High Competent	22%	8%
Strong	3%	16%

Overall, students were judged to be strongest in "global" writing issues like organization, logical structure and factual content, and weakest in the fundamentals like spelling and punctuation and using proper citation form. This is encouraging, since it should be easier to address these more mechanical issues.

Both written and oral presentations comprise a substantial part of the capstone courses, where presentations and reports in a prescribed format are required. In addition, many students write papers and give research presentations as part of honors or research projects. This number has risen over the years, and in the past 3 years a total of 63 students gave a research presentation.

Oral communication skills are assessed in the capstone courses where students give presentations and participate in design reviews. Recently, we have started to assess both the writing and speaking ability of our students in the capstones using a standard rubric. The results of our targeted assessment in the capstones are shown in the graph below. This shows that about 20% of the students were assessed as below expectations. Feedback from our industrial affiliates and our own students and faculty confirms this assessment, and that we need to spend even more effort on improving the writing and oral presentation skills of our students. We believe that this requires more practice so that communication becomes a part of the work that students perform. We have already taken steps in this direction. Many more courses incorporate writing than just 5 years ago, and this writing is being incorporated into the coursework, not just added on as separate



assignments. We recently invited Professor John Webster, the College of Arts & Sciences Director of Writing and University Coordinator for the Puget Sound Writing Project to give a presentation to the faculty on how writing can be most effectively incorporated into courses. We are now starting to use standard assessment rubrics for written and oral communication in the capstone courses.

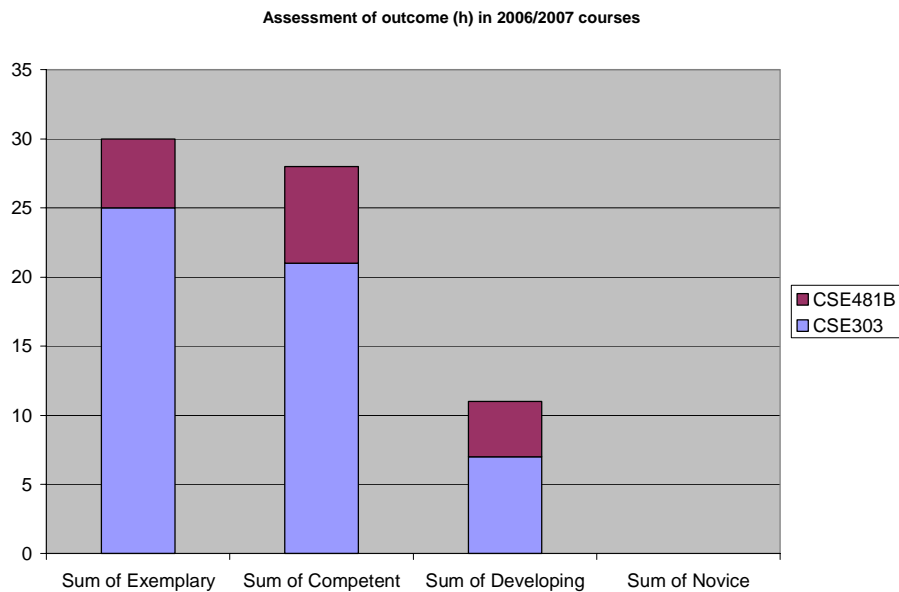
h) The broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context.

Our students gain a broad education in a number of ways that includes courses in other departments in the University, as well as experiences outside the University such as co-ops and internships. All students take a set of required courses in the humanities, which includes courses in history, philosophy, religion and psychology. Students are also required to take courses in the Arts

which includes English, foreign languages, music and drama. In addition, students have the opportunity to take a wide range of courses as free electives.

Experience outside the university in co-op and internship experiences also add to this broad education by giving them direct experience with application of engineering solutions in real contexts.

We assess this outcome for students in CSE 303 (Software Practices) as well as in the capstone courses. The results of our initial assessment are shown below. Although our assessment data is somewhat limited thus far, it does indicate that our students are meeting this outcome.



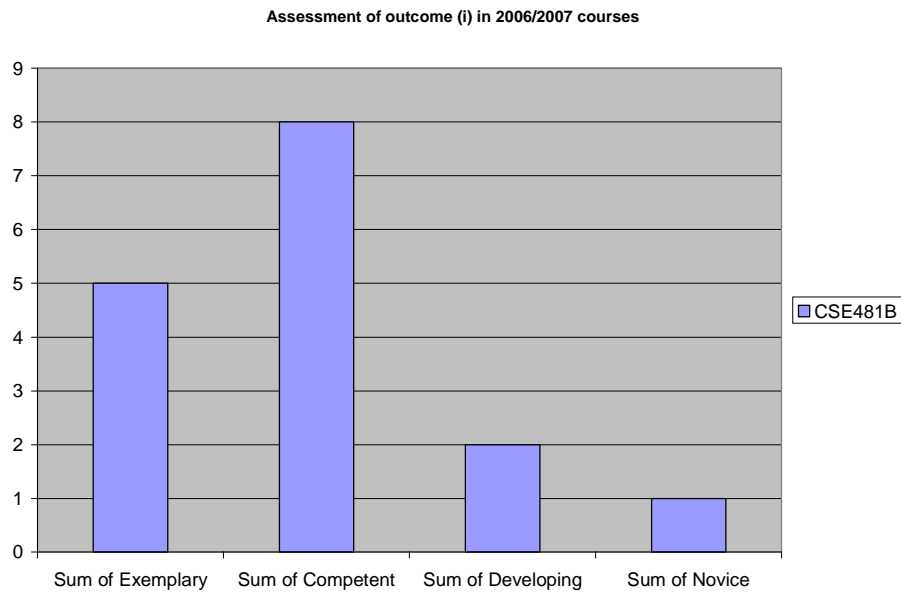
i) A recognition of the need for, and an ability to engage in life-long learning.

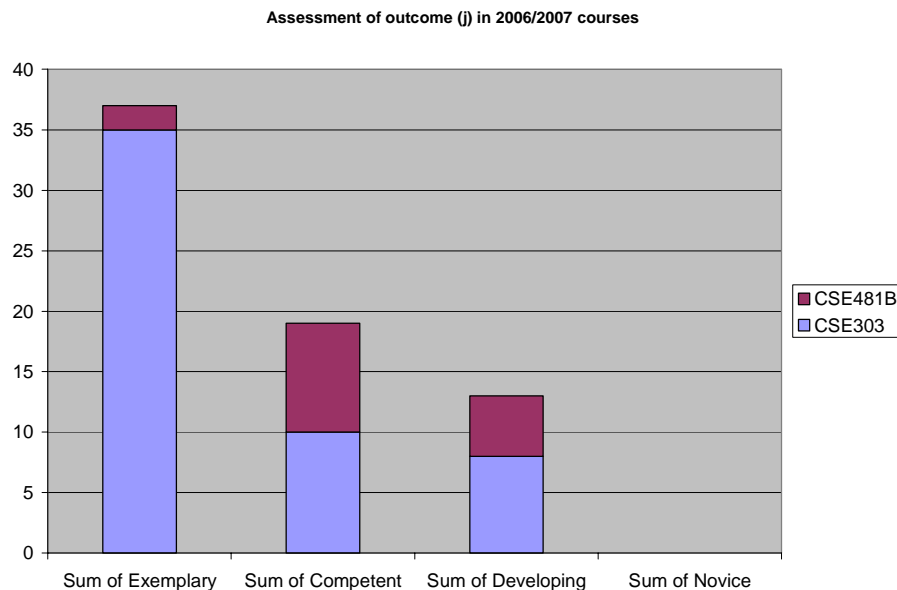
We consider the range of opportunities our students have for learning outside the classroom as one of the real strengths of our program. Most of our students take advantage of one or more of these experiences, which expose students to the range of prospects they will have after graduation, including industry practice, research and further education. Of the 140 students in the 2005 graduating class, 118 (84%) had either a coop or internship experience, or both. In addition, large numbers participate in independent research projects or as members of research teams on faculty and graduate student supervised research projects. Of the 160 students in the 2006 graduating class, 72 (45%) have earned credit for independent research with faculty. Our stature as a top research department means that undergraduate students are exposed to ideas on the cutting edge, and how advanced study can involve them in working at the leading edge of engineering and technology.

Capstones require students to apply the concepts learned in previous classes, but also to learn new concepts and tools necessary to complete their specific project. This experience requires students to extend their knowledge in a self-directed way to solve new problems with new concepts and tools. This ranges from learning new programming tools, new design and simulation tools, to finding new components and understanding the datasheets to use them, to finding other

students and faculty that can help them understand a new concept that they need. In addition, students achieve this outcome when pursuing independent and research projects.

As such, the capstone projects act as our primary and most comprehensive evaluative tool. The success rate on capstone projects demonstrates that students achieve this outcome. However, we are working on implementing a more quantitative measure of this outcome, using a rubric in the capstone courses to document and assess the extent to which students extend their knowledge of tools and concepts. The result our initial assessment of this outcome is shown in the graph below. Although the data is limited, it does indicate that students are meeting this outcome. This is also confirmed by the student self-assessment in the capstone interviews.





j) Knowledge of contemporary issues.

Students have many opportunities to interact closely with industry practitioners. Over 80% of our students participate in internships, coops and summer jobs at companies where they work closely with engineers in the industry. Fortunately for our students, they are in high demand even before they graduate and so have many different options for discovering what it is like working in industry and making contacts with possible future employers. The department and the ACM student chapter also hold events that bring students in contact with industry representatives. These include talks, job fairs, recruiting dinners, and field trips.

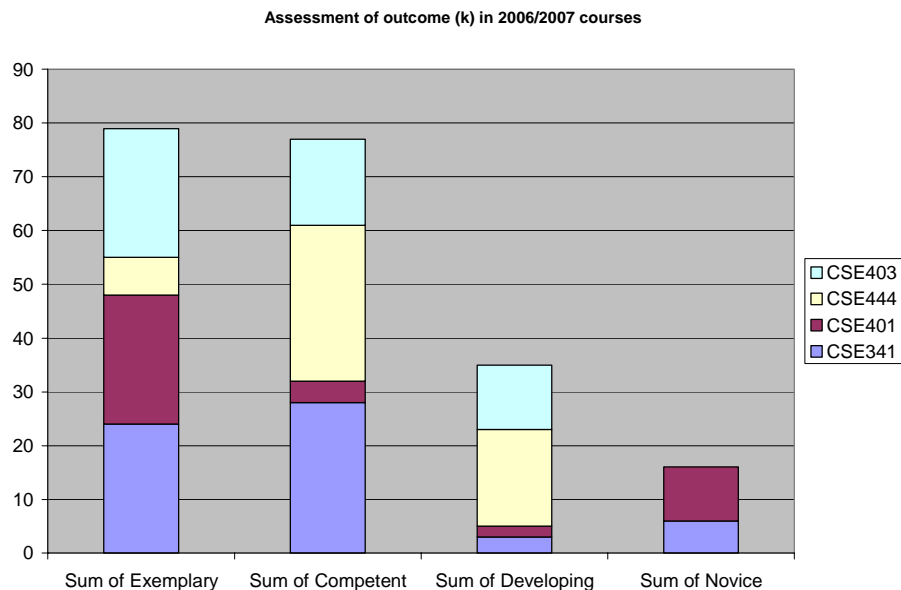
We have targeted assessment of this outcome in CSE303 and in the capstones. The limited data we have gathered thus far indicates that our students meet this outcome. Students in the capstone interviews generally agree with this assessment. We have also assessed this outcome in part by using the participation rate of students in coops and internships, by tracking the on-campus industry events, and by using student exit surveys. Our evaluation is that most of our students achieve this outcome, and that we provide a large range of opportunities for students to meet with industry practitioners.

k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

A real strength of our program is the incorporation of a wide variety of “industrial-strength” tools in our courses. These tools include a wide range of programming languages and debugging tools for both desktop and embedded systems, integrated design environments (IDE) like Eclipse, and modern hardware system design tools for schematic capture, simulation and synthesis from Aldec, Xilinx, Synplicity and Synopsys, VLSI design tools from Cadence. These tools are used over and over again for assignments and student projects.

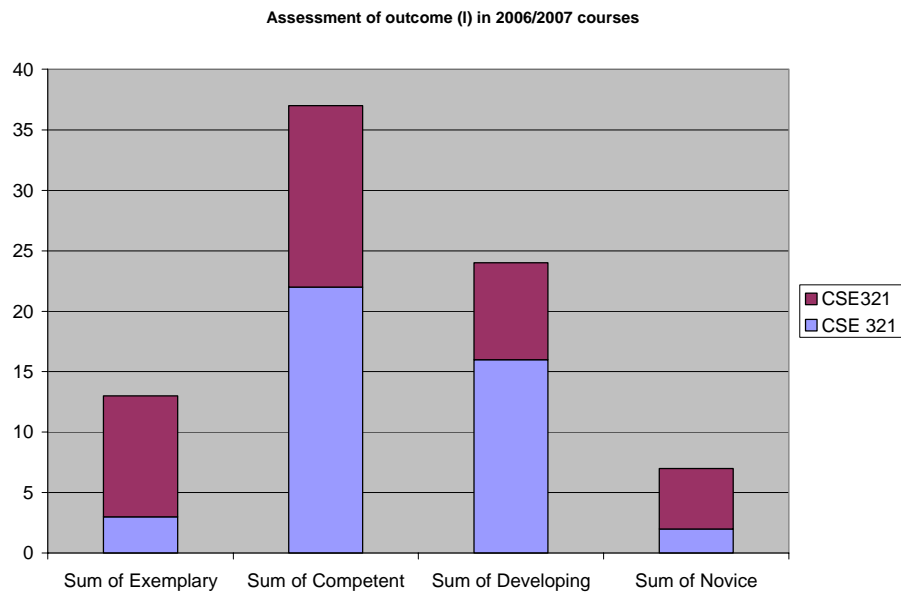
The course assignments, project and the capstone design projects gives students substantial experience applying the techniques and skills required for modern engineering practice. Successful completion of these projects requires our students to be competent in the outcome, and thus our primary assessment of this outcome is performed by instructors in project and capstone courses. These courses are CSE466 (Embedded Systems), CSE467 (Advanced Digital Design), CSE403 (Software Engineering), CSE421 (Algorithms), CSE444 (Databases), CSE471 (Computer Design) and the capstone courses. This assessment has been positive and is backed up by examples of student design work. We have begun a targeted assessment of this outcome in these classes and the initial data is shown below.

In addition, student feedback in exit interviews and surveys, and alumni surveys consistently gives high marks for their exposure to a range of tools and their ability to use them. Our affiliates also tell us that our students are very well prepared with the skills for engineering design.



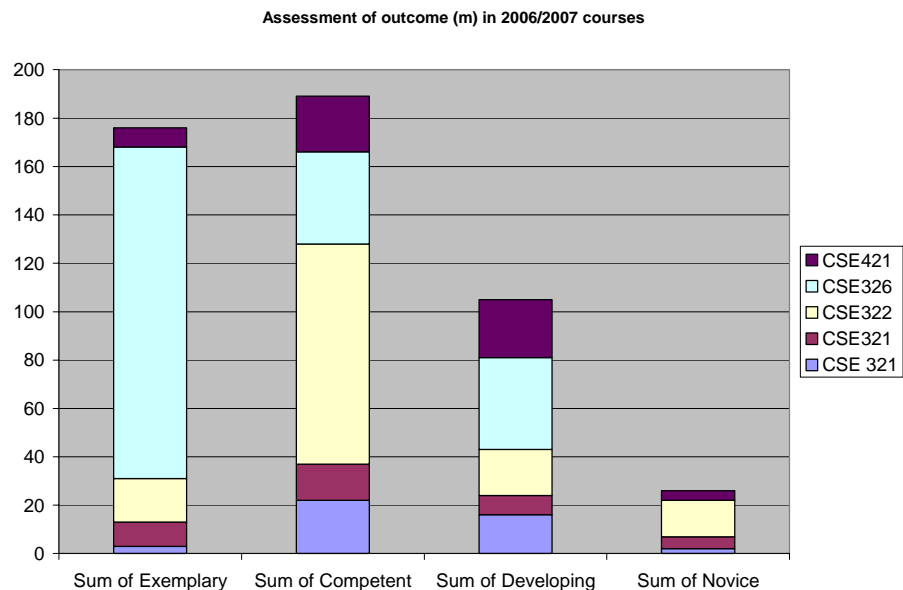
(l) knowledge of probability and statistics

Students obtain a strong background in probability and statistics in the STAT 390 (Probability and Statistics) course. Discrete probability theory is also covered in CSE321 (Discrete Structures). We then perform a targeted assessment their ability in this outcome in CSE321. The first results of this assessment are shown in the graph on the next page. Generally this assessment indicates that students are meeting this outcome. However, since students often take CSE 321 before they take STAT 390/391, we are probably assessing students too early for this outcome.



(m) knowledge of discrete mathematics

Students learn and use concepts in discrete mathematics in a range of courses in the curriculum. CSE321 (Discrete Structures) is focused primarily on discrete mathematics and covers a range of topics including logic and Boolean algebra, set theory, proofs, induction and recursive functions and graph theory. These concepts are then extended and put in practice in courses like CSE322 (Introduction to Formal Methods), CSE370 (Data Structures) and CSE421 (Algorithms). We assess this outcome formally in CSE321, CSE322, CSE326 and CSE421. We show below the assessment data that we have collected thus far. This shows that a large number of students are performing at a high level of competence and very few are performing below expectations.



Program Evaluation and Improvements

We have made many changes to improve our program since our last ABET review. The largest and most far-reaching are described in detail below. The many other smaller changes that we have made as part of our continuous evaluation and improvement are summarized in the Table at the end of this section. This will highlight many of our evaluation and improvement feedback loops in action, and show how we continually monitor the program and take action quickly to address issues that arise, or to address weaknesses in the curriculum. Some of these evaluation and improvement loops are tight, happening continually on at most a quarter by quarter basis. For example, the Undergraduate Program Coordinator meets with the advising staff each week, and issues raised by students or faculty are addressed immediately. Courses are evaluated every quarter via student teaching evaluation and teacher self-evaluations, with improvements taking place quickly if needed. Other loops are larger and take place over a longer time. For example, end of program interviews often reveal more systemic problems that require larger changes. For example, a new course, CSE 303 (Concepts and Tools for Software Development), was introduced in 2004 after evaluations by both students and instructors indicated that students were not well-prepared to handle the projects in upper-level courses.

Curriculum Revision

The largest change to our program was a revision of the computer engineering curriculum to increase flexibility and respond to specific problems that students and faculty reported. In 1999, we differentiated our computer engineering program into two different areas of emphasis, hardware systems and software systems. This was done in response to our assessment of the needs of our constituents, particularly in Washington State, which has a very large software industry. This change was very successful, attracting many new students who saw the benefits of a computer engineering degree with a concentration on software systems.

However, student surveys and faculty evaluations indicated that the program needed some fine tuning. Feedback from the admissions process indicated that students were choosing between computer science and computer engineering based on secondary and often irrelevant factors like the set of courses required for admissions. Students in the software track wanted more flexibility to take advantage of new courses. Student surveys and an evaluation of course material indicated that a more appropriate set of Electrical Engineering courses would better serve the needs of the hardware track students. Finally, a review of the hardware courses determined that improvements needed to be made to increase the coverage of a combination of hardware and embedded systems design.

A committee comprising several Computer Engineering faculty was formed to recommend changes to the program. These recommendations were discussed and approved by the full faculty at the annual faculty retreat. These changes to the program are summarized below:

1. Prerequisite courses for admission were changed to be as similar as possible for computer science and computer engineering.
2. EE 331 (Devices and Circuits) was removed as a requirement from the program, and EE 233 (Circuit Theory) was removed from the software track.
3. CSE 466 was made an elective for the software track and added as a required course for the hardware track.

4. Math 307 (Differential Equations) was removed as a requirement.
5. Several courses were added as electives to increase flexibility.

In addition, we made several changes to improve the quality of the courses and make computer engineering more attractive.

1. We added a laboratory section to CSE 370 (Introduction to Digital Design), which replaced the recitation sections. This has increased the motivation of students in this class and has given students a much better design experience.
2. We assigned instructors whose research was related to hardware design to teach CSE 370. This has increased student satisfaction with the course, and, along with the new lab, is giving students a better background in digital design.

Changes to Introductory Programming Courses

A second major change to our program was the switch of Introductory Programming courses (CSE 142/143) from C/C++ to Java. This change reflected the evaluation by our faculty that Java is a better choice for teaching modern programming language concepts. We subsequently modified our approach to teaching Java based on our evaluation of student performance in CSE 142/143 and feedback from other departments whose students take our introductory programming courses. We have moved away from an “Objects First” approach and now use a more traditional approach which introduces students first to imperative programming before moving to more abstract concepts like object-oriented programming and inheritance. This allows students who will be working in other fields to master the concepts that transfer well to other languages commonly used in engineering like C and Matlab. Our evaluation of these changes to the intro courses is extremely positive. Course evaluations have improved markedly from an average of 3.6-3.8 several years ago to between 4.0-4.2 now, which is in line with other courses in the department. We have also seen an increased interest by students in the computer science and computer engineering degree, particularly from students who have had little computing background.

CSE 303 – Concepts and Tools for Software Development

Some time after we changed from C/C++ Java in the Introductory Programming courses, we began to find that students in the higher level systems courses like Compilers, Networks, Operating Systems and Embedded Systems were less prepared to tackle projects that required substantial programming at the system level. We had assumed that students would be able pick up the low-level programming concepts and tools on their own or via the seminars run by the local ACM chapter. It was clear that this missing background was especially crucial to the success of the computer engineering students.

In response, we designed a new course CSE 303, Concepts and Tools for Software Development, that covers a wide range of low-level programming concepts as well as software development tools such as debuggers, IDEs, version control systems, and make systems that are essential for the design of large systems. We also took the opportunity to bolster other areas of the curriculum that were a bit “light”. Thus the syllabus for CSE 303 includes coverage of a number of areas where computer systems interact with society and raise ethical issues, including privacy and security, software reliability, intellectual property, and electronic voting. CSE 303 is now a required course for all computer science and computer engineering students.

New Courses

We have greatly extended the breadth of course offerings over the past several years to introduce topics like Security, Robotics, and HCI, which are becoming increasingly important in computer engineering, as well as to extend our capstone course offerings to cover a broad range of topic areas.

- CSE 454 – Advanced Internet Systems Capstone
- CSE 455 – Computer Vision
- CSE 459 – Computer Animation Production
- CSE 460 – Computer Animation
- CSE 481A – Operating Systems Capstone
- CSE 481B – Tablet Computing Capstone
- CSE 481C – Robotics Capstone
- CSE 481E – Urban Simulation Capstone
- CSE 481 F – Computational Biology Capstone
- CSE 490G – Data Compression
- CSE 490C – Introduction to Computational Biology
- CSE 490I – Neurobotics
- CSE 490K – Security
- CSE 490JL – Human-Computer Interaction

The table on the following pages summarizes other changes that we have made to the program. This table illustrates the processes at work in the department to assess our success fulfilling our educational objectives and outcomes, and the feedback loops that bring problems to the attention of the appropriate staff and faculty for action. There are many cases of the advisors and/or the Undergraduate Program Coordinator addressing issues that do not appear in this table. For example, the advisors often hear from students about problems taking place in a course. In these cases, the Undergraduate Program Coordinator will discuss the problem with the instructor and suggest remedies as necessary. These may include reassignment of a TA, additional TA help, revised project expectations or asking CIDR for a course evaluation.

Timeframe	Concern	Source of Information	Response	Results
2002-2003	Decreasing interest and enrollments in Computer Engineering	Class enrollments, student feedback, course evaluations	Committee formed to propose program changes for review and action by the department.	Substantial revamping of the computer engineering curriculum leading to increased enrollments in computer engineering, especially the hardware track.
2002	Lack of agreement in admissions requirements for computer science and computer engineering degrees.	Program review, student feedback	Changes to make the prerequisite courses for admission to the computer science and computer engineering programs as similar as possible	Allows students for CS and CE to apply at the same time in their career and a reduced barrier to choosing CE major. Increased flexibility in choosing CS and CE majors.
2002	Lack of appropriate EE background for computer engineering students	Faculty evaluation of student abilities, student self-assessment and comments.	Change in program prerequisites. Discussions with EE faculty to adjust coverage of topics in EE courses.	Increased student satisfaction with EE course material.
2001	Lack of access to advanced HW platforms	Faculty evaluation	Solicitation of major donation of FPGAs from Xilinx, and purchase of design platforms for labs.	More interesting and complex projects in the advanced design courses. Increased student interest.
2002-2003	Lack of interest and falling enrollment in the HW track.	Course enrollments, student feedback, Faculty evaluations	Introduction of laboratory section in CSE370, replacing recitation section. Assignment of front-line computer engineering faculty to CSE 370.	Greatly increased student interest and satisfaction with course. Better student and faculty course evaluations.
2002	Outdated CAD design tools	Faculty course evaluations, program review	Survey of available design tools and adoption of Aldec HDL for HW design, and Synplicity for synthesis.	Tools provide integrated schematic and Verilog design for smooth transition from beginning design through advanced systems design.
2003	Lack of synchronization of material in 370/378 resulting in substantial overlap	Student course evaluations, faculty course reviews.	Meetings of faculty concerned, 370 changed to a prerequisite for 378, allowing for close coordination of material. Codification of simple processor design introduced in 370.	Very positive reaction from students, elimination of overlap in courses, allowing coverage of more concepts in 378.
2003	"Stale" platforms and projects in the embedded systems course CSE 466	Faculty course evaluations	Upgrade of 466 to using state-of-the-art sensor nodes.	Leading edge sensor net projects in CSE 466 and 477. Results published in Computer.

2002-2006	Importance of capstone design experience to computer engineering outcomes	Feedback from Affiliates and alumni	Increased number and variety of capstone courses	Increased computer engineering enrollments, increased relevance to student career choices
2003	CSE 461 (Networks) did not have enough practical and active learning and too much "book work"	Faculty course evaluations	Added one credit hour to CSE461, allowing a revised course structure to include a project, making the course more design-oriented and less rote learning.	Improved student evaluations, improved faculty course evaluations
2003	Importance of statistics to computer engineering	Faculty input	Introduction of STAT 391 – Statistics for computer science and engineering students	
2004	Project in CSE401 (Compilers) was out-of-date, and too focused on parsing and too little on optimization and code generation	Faculty course evaluations	Complete revamped project component	
2004	Lack of "theory into practice" experience in CSE 473 (Artificial Intelligence)	Faculty course evaluations	Added group projects in CSE 473.	
2003	Lack of coverage in curriculum of modern processor design concepts.	Faculty program and course evaluation	Redesigned CSE471 (Computer Design) to give students a broad understanding of the most advanced concepts used in current and future processors	Improved faculty course evaluation, strong student course evaluations.
2001-2004	Lack of exposure to professional ethical issues	Previous ABET self-study, student feedback, program review	Inclusion of coverage of ethical issues in CSE 303	Increased awareness by students of ethical issues in the practice of computer engineering
2004	A decrease in student background in the concepts and tools required to complete projects in senior systems classes.	Instructor course evaluations, student assessment and feedback. Program review of effects caused by moving the introductory programming class from C/C++ to Java.	Introduction of new course, CSE 303, to cover practical aspects of building large programs and systems, focusing on a broad range of programming and software engineering tools.	Increased ability by students to construct large projects and complete assignments in advanced computer engineering courses.

2004-2005	Decreased ability of students to program effectively	Faculty evaluation of student programming abilities, program review, feedback from Affiliates and other COE departments	Changes to the introductory programming sequence to focus on the core concepts and abilities required for designing and implementing algorithms, and less on object-oriented, large system programming. These concepts are covered later in the program, including CSE341 and CSE303.	Increased programming proficiency of students taking 142/143 sequence.
2002-2004	Decreasing interest in computer science and engineering	Admissions statistics, student evaluations	Addition of women's seminar and honor's seminar for 142/143 students.	Stable admissions statistics and student population
2005-2006	Lack of consistency between offerings of the same course	End-of-program interviews and meetings with advisor and chair	Assignment of faculty course supervisor	Reduced complaints from students
2005	A large variability in undergraduate TA abilities	Student evaluations, end-of-program interviews, faculty evaluations.	Added an interview process for choosing undergraduate TAs, instituted training for all TAs and weekly TA "community meetings"	Greatly reduced student complaints, improved faculty reviews.

4 Professional Component

The curriculum is structured to provide both depth and breadth. Students majoring in Computer Engineering have the opportunity to choose a hardware-focused or software-focused program. A set of common core courses forms a foundation on which senior requirements are layered, senior-elective requirements ensure breadth outside the area of specialization, and appropriate capstone design courses for the two options provide an in-depth design experience. Our courses are structured to ensure that all students can achieve all the outcomes that we have defined for the program.

Our curriculum requirements are grouped into 9 categories:

- General education
- Visual, literary, and performing arts/Individuals & societies
- Written and oral communication
- Mathematics and statistics
- Science
- CSE core
- CSE hardware or software option outer core
- CSE electives
- Free electives

General education and VLP/IS requirements expose our students to a variety of topics in the humanities and the natural world. Written and oral communication requirements include two technical writing courses in addition to basic English composition. Mathematics includes one-year of calculus, and courses on linear algebra and statistics. The science requirement includes two courses of Physics and 10 more credits of Natural Science. Students can further increase their range using their free electives. Many of our students pursue a foreign language or a double major using this mechanism. Many use these credits for additional independent study in the department (6 credits of which can be counted in CSE electives, 9 credits if part of the Department or College Honors program) or to take more electives within the department.

The heart of our curricular requirements is the departmental core requirements which are arranged in three tiers. First, and common for all our students in both Computer Engineering and Computer Science, are the common core requirements that ensure a solid foundation in the discipline. This component consists of 9 required courses. Computer engineering students choose between two options for the outer core requirements: one focusing on hardware systems and the other on software systems. Each option has three required senior level courses. In addition, a capstone course is required that ties together many of the disparate elements of the curriculum and involve the students in a large implementation project where they work in teams and are exposed to complex engineering tradeoffs. The CSE electives serve to further ensure breadth in the discipline and can include credit for research participation.

The Computer Engineering program curriculum is described in detail in the department undergraduate programs handbook can be found on the web at: <http://www.cs.washington.edu/education/ugrad-brochure/handbook.html> . The curriculum is designed by faculty as a whole through the Curriculum Committee and the Undergraduate Program Coordinator. After changes are ratified by the faculty, they also must receive approval by the College's Educational Policy Committee and the University's Curriculum Committee. The

handbook, the tables above listing our courses and their mapping to our objectives, and a detailed curriculum plan outlined in Appendix I provide all the required information. A summary sheet that contains all the computer engineering program requirements is given in Appendix 1D.

In summary, the program consists of the following requirements based on a 180 credit degree program under a quarter system. There are two options that emphasize either hardware or software.

Core requirements (35 credits): CSE 142, 143, 303, 321, 322, 326, 341, 370, 378

Additional Computer Engineering requirements (12 credits): CSE 451, 461, and EE 215

Hardware Specialization (18 credits): EE 233, CSE 466, and 467;
Hardware capstone: CSE 477

Software Specialization (15-17 credits): CSE 403, and two of CSE 401, 421, 444, 466 or 471;
Software capstone: CSE454, CSE481

CSE Electives (6-7 credits): all 400-level CSE courses not listed above

Math (19-22 credits): MATH 124, 125, 126, 308, STAT 390

Science (20 credits): PHYS 121, 122; 10 credits of Natural Sciences

VLPA and I&S (30 credits): from UW provided list.

Written and Oral Communication (12 credits): English Composition, TC 231, 333

Free Electives (20-25 credits: depending on option chosen)

Table I-1 in Appendix 1 describes how these credits are allocated to Math and Science and Engineering and where substantial design takes place in the curriculum. We recently discovered a problem in our accounting of the Engineering credits for the software track of our program. A clerical error in the table assigned one course 4 credits instead of 3 and a discrepancy between the software and hardware track went unnoticed. The result is that the Engineering credits for the required courses for the software track total 66, 1.5 credits short of the 67.5 threshold (3/8 of 180).

We have already addressed this problem by changing the requirements in our curriculum to include one more course in both the hardware and software tracks. This change has been approved by the College and will take effect in 2007. The new requirements are reflected in a revised Table I-1 in Appendix 1D, and in the new course requirements document also given in Appendix D. We have conducted a credit audit for all students graduating the last 2 years. We found that almost all students have many more than 67.5 Engineering credits. However, we did find 2 students that did fall short by .5 or 1.5 credits. A complete audit of students affected by the current requirements will be available to the evaluation team during the on-site visit.

Design Experience

Students gain significant design experience throughout the program starting with relatively small design problems in the core course, to substantial quarter-long team projects in the capstone courses. Many of the required senior-level courses are taught in the context of a substantial design project experience.

CSE378 – Computer organization: The project in CSE378 begins by having students construct a MIPS based processor with a 5 stage pipeline design from a simple datapath. Starting with a basic datapath, students first construct a single-cycle processor that can execute common MIPS instructions dealing with arithmetic, branch, jump, and memory operations. This single-cycle processor is then further modified into a fully functional 5-stage pipeline to improve performance and illustrate the concept of pipelining. Finally, students take their pipelined processors and construct several parts of a memory system to allow two of these pipelined processors to maintain coherency across caches while reading from and writing to a shared memory. The end product is a dual-core architecture with coherent caches that is implemented in real hardware on an FPGA. Throughout this process, students are introduced to the details of constructing a processor not available in higher level simulation based approaches including a basic notion of the costs and benefits of various designs in real hardware. Furthermore, students are given an opportunity to experience making engineering design decisions in the final lab as they are required to construct their own specification for a communications protocol between memory components and to implement this protocol both in simulation and on an actual FPGA.

CSE401 – Introduction to Compiler Construction: This has always been an implementation class in our department. The class implements new features in a scaled down, but running, mini-compiler. This differs from the build everything from scratch approach and is preferred by UW faculty because it models the typical industry situation of working with someone else's code. Students must "get their brain around" the compiler's structure before writing their own code. Further, it shows them good practice, as the mini-compiler is always beautifully written. Students extend the language by adding statements and operations, and extend the compilation by improving code generation, etc. They touch all phases from lexing to target code generation. They use tools CVS (or more recently, SVN), lex-generator, parse-generator, and development tools like debuggers. Of course, along with all of the implementation, concepts, theory and best practices are covered in lecture. Past graduates have commented years after leaving campus that it was one of perhaps two most important classes for preparing them for a development career.

CSE403 – Software Engineering: Students complete a 9-week design and implementation project. The students propose ideas for software systems to build and are placed into groups of 5-8. Several deliverables are required during the course of the projects, including a requirements analysis, design specification, beta and final code releases, and testing materials. The design specification includes the submission of several UML diagrams such as a class diagram and several sequence diagrams. The students are taught and asked to use modern tools such as content management systems, UML diagram generators, and testing frameworks to generate their materials.

Examples of engineering design decisions students were required to make included:

- What features should the project have? What is the relative importance of these features, i.e., which ones should be cut if time becomes tight?
- What languages and tools should we use to design and implement these features? This includes CASE tools such as UML editors, schedulers, version control, bug tracking tools, as well as the overall programming language(s), APIs / libraries, and IDEs used to write the code.
- What should be the testing plan? They had to do unit testing as well as system testing of some form.

CSE451 – Operating Systems: The objectives of the project component of CSE 451 are to a) provide experience in working in teams to design, implement, and evaluate important components of modern operating systems, b) provide experience in recognizing, evaluating, and making the engineering design tradeoffs that characterize operating system design and, in fact, many other aspects of computer system design, c) provide experience in conducting performance experiments and understanding the performance tradeoffs that are crucial in operating system design and, again, in many other aspects of computer system design, and d) provide experience in producing coherent written technical descriptions and assessments of designs, design tradeoffs, and implementations.

The first project assignment involves adding a system call to the Unix kernel and building a rudimentary Unix shell. This project provides detailed knowledge of the design of the Unix shell, experience with confronting a large, complex software artifact such as the Unix kernel, experience compiling and installing modified operating system kernels, and teaches the ability to make small but critical changes to the operating system.

The second project assignment involves building a user-level thread package running on top of Unix, building a web server that utilizes this thread package, and conducting performance experiments to understand the costs and benefits of concurrency. This project provides detailed knowledge of concurrency, concurrent programming, thread management, and performance evaluation. This project has a significant writing component which is evaluated separately.

The third project assignment involves enhancing a bare bones Unix file system implementation in various ways -- for example, to support more files, larger files, or longer file names. The file system is one of the most complex and interesting components of a modern operating system, and is characterized by a large number of interesting engineering tradeoffs between complexity, functionality, and performance; this project provides detailed experience with all of these, and gives the students direct experience with using a standard extensibility interface (the VFS layer) that exists in modern kernels.

Feedback from our alumni and their employers consistently indicates that this course is one of the most beneficial in our curriculum in terms of preparing students for engineering careers where they will tackle these sorts of issues on a regular basis.

CSE 461 – Introduction to Computer-Communications Networks: A recent course project was to design and evaluate an algorithm for identifying RFID tag identifiers: a large number of tags are assumed to be within range of the reader for only a short period of time, the communication channel is noisy, and the goal is to identify as many tags as possible (while also controlling false positives). Students were free to choose evaluation either through testbed implementation, using a locally built RFID simulator, or analytic models. The "deliverable" for the project was a set of design documents and algorithm evaluations, prepared incrementally as the students were doing their work, and submitted through a web form interface. Each document had four sections: major version description, minor version description, evaluation results, and conclusion. Their purposes, respectively, were to give a high level overview of an approach, to discuss specific decisions made within the general approach, to present evaluation results, and finally to analyze those results to both identify weaknesses of the current scheme and to postulate how to address those weaknesses. The students would then implement the proposed changes, evaluate it, and prepare a new report. A broad view of the problem was taken: as well as designing an algorithm that works within the current EPC Class-1 Generation-2 RFID protocol specification, they were allowed to suggest and evaluate reasonable changes to that specification motivated by the results of their experiments. Additionally, extra-credit extensions were suggested (e.g., what if there are two

readers rather than just one?). Student design iteration ranged from one major version/three minor versions to as many as three major versions (with many minor versions).

(Project writeup:

<http://www.cs.washington.edu/education/courses/461/06au/homework/hw3b.html>)

CSE466 – Embedded Systems: The design project for this class is built around the UC Berkeley Mote sensing and communication platform. An example project is the “Flock Project”, which incorporates the major embedded systems and wireless communications concepts and techniques. The Flock includes exposure to an embedded operating system, priorities and constrained resources, sensing, actuation, interfacing, and aspects of wireless networks, among others. Each student designs their own *bird*, using a common rule set. These birds then interact via wireless links to display emergent behavior reminiscent of a *flock of birds*.

The Flock builds on concepts presented early in the quarter, allowing students to begin writing reusable code to optimize their effort. The Flock requires that each pair of students complete a full bird implementation that interacts correctly with the others birds. This contrasts with senior Capstone projects where each student completes a portion of a larger project. In addition, the use of a test fixture for qualification teaches students a valuable lesson in completeness of specifications and testing. The project is partially open-ended which allows students to explore both the design and solution space.

CSE467 – Advanced Digital Design: Students develop a full implementation of a complete hardware system that implements some real-time function like a graphics board or a video-rate image processor. In the image processing project, students started out by developing a video pipeline one component at a time. This comprised an interface to the video codec, followed by components that performed timing decoding, color space conversion and de-interlacing. Once the complete pipeline was complete, students then had the opportunity to implement a variety of real-time image transforms including scaling and rotation with bi-linear interpolation. This requires students to manage the complexity of a fairly large system using concepts of hierarchy, modularity and interface design. Students also gain substantial experience with state-of-the-art design tools for specification, simulation and synthesis, as well as modern FPGA architectures and hardware platforms. Our affiliates tell us that our students coming from this hardware design experience, complemented by the capstone course, have both the background and skills to become productive engineers almost immediately.

Capstone Design Experience

The capstone design experience gives students the opportunity to construct a more substantial project than a course project by allocating more credit hours, tying together concepts from several courses, and having students work together in larger groups, typically with 3-4 students. These courses take place at the end of the program for the students and are a place where students get a chance to show off all that they have learned.

Over the years we have introduced a variety of capstone courses that focus on different aspects of computer engineering. We also describe here some of the capstone projects that have been done in the past few years. These projects are often presented to the larger department community, including colleagues and industrial affiliates, in the form of live demonstrations, videotaped productions, and documentation published on the Web. The project reports and artifacts for representative design projects will be available for the ABET evaluation team.

CSE 477 Hardware Capstone: This course has a strong focus on hardware and embedded systems design. Projects range from hardware-intensive projects that are implemented using large FPGA platforms and hardware description languages like Verilog to embedded systems projects based on distributed embedded sensor platforms. Most projects comprise a combination of hardware and software components and leverage the different expertise of the project team members. CSE 477 is required of all hardware track students, although some software track students also take it.

CSE 454 Advanced Internet Systems Capstone: Students combine techniques in basic Internet programming with concepts from information retrieval, machine learning and data-mining to construct scalable and secure Web-based services for searching and classifying Web-based content. In one example project, students constructed a large-scale Google-style search engine with a focus on using data-mining techniques to provide high-quality ranking for large numbers of search results.

CSE 481 Computer Games Capstone: This course is centered on giving students experience with hands-on design and development of a substantial software product. In the capstone project, students learn how to use modern, real-world tools to build complex software systems. The games domain requires students to deal with the complexity and performance demands of high-performance, real-time, distributed application. Student teams implement games of their own design, so that they are involved with the creative process from start to finish. At the end of the course, students present and demonstrate their work to a larger department audience.

CSE 481 Operating Systems Capstone: Students build on the fundamental computer operating system concepts learned in the introductory course (CSE 451) and explore how those concepts are actually incorporated into the Windows operating system. Students then work in teams to add extensions to the Windows operating system using Windows 2000 sources, which have been provided by Microsoft to the UW under a Nondisclosure Agreement (NDA). Students propose their own projects, which are approved by the instructor. Some recent projects have included;

1. Designing and extending the DOS FAT file system to support per file compression. This required extending the on-disk format of the file system and altering the file system to support the new feature.
2. Redesigning the Windows scheduler to be less thread-centric to fairly schedule threads among processes/jobs. Currently in any windows system the kernel schedules threads to run regardless of which process/job they belong to. This means that a process with 100 threads can quite easily starve a process with 5 threads. The project is to make scheduling more equitable.
3. Understand and fix a problem in Windows where it wrongly evicts large processes from memory. In windows, not including Vista, any program that scanned files on the disk, such as virus scanners, caused this eviction to happen. The project is to understand and remove this behavior.
4. Add rootkit detection to the windows kernel.

CSE481B – Tablet PC Capstone: The main goal of this course is exploratory development using the Tablet PC platform, culminating in the production of an innovative application that takes advantage of the Tablet PC. Class sessions are divided between regular lectures at the beginning of the quarter, guest lectures, student presentations, and group meetings. Students work in teams to develop projects that are based on pen-based computation. Projects are evaluated on criteria such as Team Work, Engineering Quality, Release Quality, Usability, Design Process, Involvement of Potential Users, etc. Example capstone projects have included:

1. Note-taking support: This provides services that increase the accessibility and usability of the Tablet PC note taking abilities by automating the uploading of notes taken on a tablet to a centralized server where the notes are converted to a web readable form.

2. Handwriting generation: One team developed a handwriting generation program (the opposite of handwriting recognition) - where a user inputs text and the output is cursive handwriting based on samples taken from the user.
3. Editor of handwritten text: The handwriting editor allows editing of handwritten documents (as Word allows editing of text documents) – for example, if a word is deleted, the handwritten words re-flow on the page.
4. Diagram recognition: The diagram recognition project focused on recognizing handwritten diagrams representing basic data structures - such as trees, lists and arrays.

CSE481C – Robotics Capstone: This capstone covers concepts in Robotics, culminating in a capstone project that applies these concepts to a sizeable problem. These concepts include behavior-based techniques, probabilistic techniques, particle filters, Markov design processes and active sensing. Each offering poses an open-ended problem that students focus on in groups. The first offering used the AIBO robot dogs and the Robocup challenge as the example problem domain. The most recent course centered on autonomous navigation using a blimp that was flown in the department's atrium. Each project group focused on a different aspect of the problem including tracking, control, simulation and visualization.

CSE481E – UrbanSim Capstone: UrbanSim is a simulation system for doing integrated modeling of land use, transportation, and environmental impacts to help inform public decision-making about major transportation and land use decisions. In this capstone, students interact with students from Urban Design and Planning and the Evan School of Public Affairs to design a software project in the context of UrbanSim to solve an interesting problem.

CSE490F – Computational Biology Capstone: In this capstone, students are taught the basic tools of computational biology, including molecular biology, biological sequence analysis, current computational tools and databases for computational molecular biology. They then apply their computer software engineering skills in teams to design, implement, and test a software system to perform high throughput analysis of a problem in molecular biology. In most cases, the project teams each contribute a part of the overall solution, and must co-ordinate their efforts to produce the final product. Example projects have included developing software that identifies evolutionarily conserved motifs in the DNA regulatory regions of homologous genes from multiple bacteria, phylogenetic footprinting in yeast species, and cataloguing the prokaryotic regulatory elements.

Extra-Curricular Activities and Outcomes

We also use extra-curricular elements to further accentuate outcomes that are particularly important to engineers practicing in our field or continuing on to graduate school. These are:

- Industry experience by means of co-ops and internships
- Research participation
- Teaching assistantships

Although our students are not required to participate in these extra-curricular elements, a large number of them do participate and gain the most from what our type of department has to offer. For example, more than 80% of our students are recorded as having completed industry internships. Of course, this number is likely much higher as not all students inform us of their employment although our advisors try to elicit this information when students complete their graduation application. Approximately 45% of our students participate in research with faculty and graduate students. Many of our undergraduates also publish research papers with faculty.

For example, in the 2004/2005 academic year, 24 different students published a total of 15 research papers.

Finally, over 50% (up from 10% six years ago) of our graduating class each year has participated as teaching assistants or teaching consultants not only for our introductory courses but for many of our capstone and other laboratory intensive courses as well. We find that our own undergraduates, who are familiar with our equipment and development environments, are often more effective than graduate students, who have not worked directly with the same tools.

We strive to increase participation in these co-curricular activities via information sessions run by the ACM student chapters, orientation comments and explanations by the Undergraduate Program Coordinator and Chair, through our Undergraduate Advising Unit, and through student mentoring. The participation rate has been steadily rising as students become aware of these options and talk to fellow students who have benefited from their experiences.

5 Faculty

The department operates with minimal hierarchy. Faculty of all ranks have similar teaching loads. All faculty have annual peer reviews of teaching and meetings with the Chair to discuss activity level and performance in teaching, research, and service. Annual evaluations include a reflective statement from each faculty member to the Chair outlining their assessment of the past year and goals for the future.

There are 44 faculty:

- 25 full professors
- 10 associate professors
- 5 assistant professors
- 3 senior lecturers
- 1 lecturers

The CSE faculty covers a wide range of the discipline and, collectively, has a wide range of experience in our discipline including: industry research labs, start-ups, and advanced development. Faculty members are highly visible in their respective communities with many serving as chairs of major conferences and enjoying extensive networks of colleagues on an international, as well as national, level.

Twenty seven current CSE faculty members have won [Presidential/NSF Young Investigator Awards or NSF CAREER Awards](#). Six faculty members are [ONR Young Investigator Award](#) recipients. Five hold [NSF Presidential Faculty Fellow or Presidential Early Career \(PECASE\) Awards](#). Fourteen (plus three Adjunct faculty) hold [Sloan Research Fellowships](#). Three hold [Packard Fellowships](#). Among the senior faculty are seven [Fulbright](#) recipients, two [Guggenheim](#) recipients, fifteen [Fellows of the ACM](#), nine [Fellows of the IEEE](#), two [Fellows of the International Association for Pattern Recognition](#), three [Fellows of the American Association for Artificial Intelligence](#), one [Fellow of the American Association for the Advancement of Science](#), one [Fellow of the American Academy of Arts & Sciences](#), and two (plus six Adjunct/Affiliate) [Members of the National Academies](#).

All faculty are involved with teaching in both our undergraduate programs. Five faculty members in the department are recipients of University awards: 4 are winners of the Distinguished Teaching Award (Borriello, Diorio, Ebeling, and Salesin) and 1 is winner of the Distinguished Graduate Mentor Award (Notkin). The ACM student chapter honors a faculty member (at our annual end-of-year party) with an ACM Distinguished Teaching Award decided by a student vote. Finally, the Bob

Bandes award honors the Department's top TAs based on nominations from students.

Curriculum quality is assured through faculty self-assessments of their teaching performance after each course, peer evaluations of teaching for faculty of all ranks on an annual basis, and student evaluations of each course. The self and peer evaluations touch upon not only teaching performance in specific course offerings but also consider curriculum development, development of course infrastructure, future plans for the course content and assignments, and how previously identified issues were handled.

Faculty self-evaluations: Faculty members evaluate the courses they teach by filling a self-assessment each quarter with the Department Chair. This includes not only a post-mortem examination of the faculty member's effectiveness as a teacher but also comments regarding course pre-requisites, development, infrastructure, assignments, etc. Annually, two peers evaluate each and every faculty member, regardless of rank, for teaching performance and effectiveness. Self-assessments, student evaluations, and informal observations by other faculty who attended lectures are collected and overall advice developed to help the faculty member improve their performance.

Another measure of the success of our educational program is the recognition our faculty has received for excellence in teaching. We, as a department, care deeply about teaching as evidenced by the care we take in peer evaluation of teaching, and the attention paid to curricular issues. We routinely do extremely well in terms of student evaluations of our teaching. The student teaching evaluations have been in the range of 4.0 – 4.2 for the last 5 years (on a scale of 1-5, where 4 is Very Good and 5 is Excellent). This compares very well to other departments in the University. In addition, three faculty members have received the [College of Engineering Faculty Achievement Award](#), and four have received the [University of Washington Distinguished Teaching Award](#). In 1999, CSE received the inaugural UW Brotman Award for Instructional Excellence -- in essence a departmental distinguished teaching award. In 2007, Richard Anderson received the College of Engineering award for Innovation in Education.

Our faculty continually strives to improve its teaching performance. All our courses are evaluated by students. Peer evaluations of teaching are taken very seriously. Graduate student instructors are provided with a seminar to develop their skills (CSE 590IT). The faculty takes full advantage of the resources the University and College have to offer. In particular, many of our faculty have had representatives from the Center for Instructional Development and Resources (CIDR – <http://depts.washington.edu/cidrweb/>) visit their classrooms and conduct student interviews and make suggestions for improving their lectures. Interactions with the College of Engineering's Center for Engineering Learning and Teaching (CELT – www.engr.washington.edu/~celtweb/) have included: individual faculty consultations with instructional consultants; faculty participation in CELT workshops and attendance at CELT sponsored talks; collaborations in educational research proposals; evaluation of educational projects; CSE funding of a project with the Coop office to investigate learning outcomes resulting from co-op experiences; and classroom assessment.

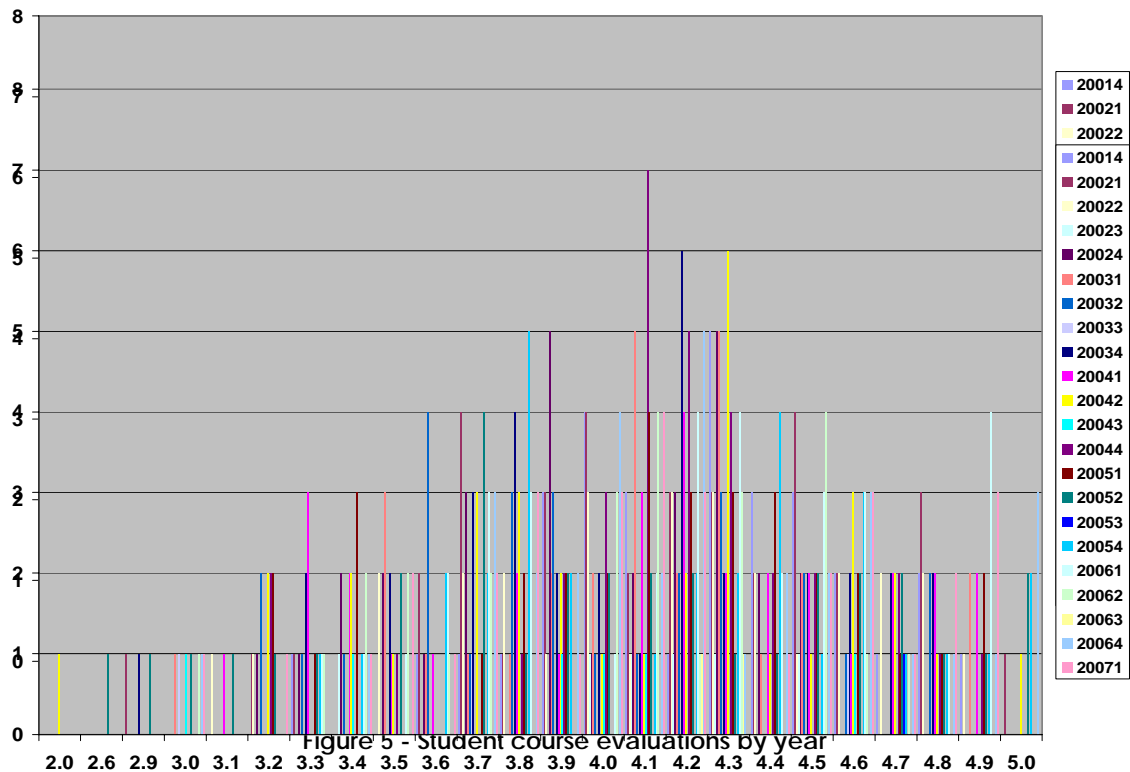


Figure 5 shows a histogram of the student course evaluations of teaching for the past 5 years). Students evaluate courses on a 5-point scale: 1=poor, 2=fair, 3=good, 4=very good, 5=excellent, on many different aspects of the class, including instructor effectiveness, organization, etc. The number reported here is the combined scores for questions 1-4, which reflects both the course and the instructor. The mean and median scores for instructors across the department have consistently stayed in the 4.0-4.2 range. (This data includes evaluations for all our courses including undergraduate, graduate and non-majors courses.)

The faculty is extremely active in research and our department is consistently ranked in the top 10 in the nation in our discipline. There are strong ties with industry with many faculty receiving research and educational support from corporations, through consulting, and past experience in industry. National visibility is high with several faculty members serving on National Science Foundation advisory boards, National Research Council study panels, and Department of Defense research organizations. One of our faculty (Snyder) is the leader in a drive to develop university and high school courses in information technology literacy (see our own CSE 100 taught in conjunction with the Information School).

Within the University, three faculty members have received the [College of Engineering Faculty Achievement Award](#). In 1996, Ed Lazowska became the first member of the College of Engineering to be named the University of Washington Annual Faculty Lecturer, and in 1998 he received the University of Washington Outstanding Public Service Award. In 1999, CSE received the inaugural UW Brotman Award for Instructional Excellence -- in essence a departmental distinguished teaching award. Eleven CSE faculty members have been recognized with [endowed positions](#).

6 Facilities

The department moved into the Paul Allen Center in 2003, greatly expanding space for labs and meeting rooms. The Paul Allen Center is adjacent to the recently-completed building for the Electrical Engineering Department and shares a large atrium that serves as a "student commons". There is a total of 85,000 square feet of space – this new building increased CSE's total space by a factor of 2.5 and its laboratory space by a factor of 3, for the first time approaching national norms for comparable programs.

A total of 11,793 square feet is devoted to lab space for general computing, graphics and animation courses, hardware design and special projects.

Instructional computing is provided through laboratories operated within the department. These include three general use laboratories with over 90 Intel Pentium PCs running Windows XP and Linux. Additional back-end resources are provided by Intel PC-based compute and file servers running Linux, accessed through "WinTel" front ends using the X Window System.

The department also operates four special-purpose laboratories containing approximately 100 Intel Pentium PCs, each backed by dedicated file servers. To support digital system design courses, the **Embedded Systems Laboratory** contains 54 Pentium workstations for design entry and simulation along with Tektronix logic analyzers, digital oscilloscopes and other test equipment.

Capstone and graduate courses in tablet computing utilize the **Tablet Computing Lab**, containing 11 HP Compaq Pentium workstations and 11 HP Compaq Tablet PCs. Computer graphics courses utilize the **Instructional Graphics Lab**, containing approximately 20 high-end Intel Pentium workstations. The **Laboratory for Animation Arts** includes 18 Intel PCs and digital video production equipment, and is used for teaching interdisciplinary courses in computer animation. The **Special Projects Lab** contains 20 Intel PCs, and is used to teach capstone courses in operating systems and other courses requiring specialized equipment or dedicated access. The SPL runs different systems and software at different times, depending on course needs.

General Computing Labs (4400 sq. ft)

- Workstation Lab (CSE 002) – 31 Windows PCs, 6 Linux PCs, 1 Multi-Media PC
- Workstation Lab (CSE 006) – 20 Windows PCs, 4 Linux PCs, 1 Multi-Media PC
- Workstation Lab (CSE 022) – 19 Windows PCs, 6 Linux PCs, 1 Multi-Media PC

Specialized Computing Labs (3922 sq. ft.)

- Tablet Computing Lab (CSE 003D) – 11 Compaq PCs, 11 Compaq Tablet PCs
- Graphics Lab (Sieg 327) – 17 Windows PCs
- Digital Animation Lab (Sieg 329) – 26 workstations, 20-node RenderFarm
- Special Projects Lab (Sieg 324) – 29 Windows PCs
- A cluster of four Unix cycle servers, and Web & database servers

Hardware Design Labs (3473 sq. ft.)

- Hardware Lab (CSE 003) – 39 Workstations
- Hardware Projects Lab (CSE 003E) – 13 Workstations

The hardware labs (CSE 003/003E) additionally contain many test instruments and hardware platforms for design of hardware projects and embedded systems. The main hardware lab (CSE 003) has 14 workbenches, each equipped with a workstation, oscilloscope, logic analyzer, function generator and power supply. The hardware projects lab (CSE 003E) has an additional 12

workbenches, 11 of which are equipped with a workstation, oscilloscope, logic analyzer, function generator and power supply. Additional test equipment like multimeters and logic programmers are also available.

A variety of hardware design platforms are used in classes. The introductory digital design course uses a breadboard attached to lights and switches, as well as an FPGA that is used for clock generation and testing. The advanced courses use a variety of FPGA-based platforms, particularly the Xilinx XUP platform that contains a 3 million gate FPGA as well as a variety of interfaces.

Complete information about the hardware lab facilities is available online at:
<http://www.cs.washington.edu/lab/facilities/hwlab/>

Design Tools for Hardware Design

A large variety of software packages are used by the hardware oriented courses:

- Aldec HDL schematic and Verilog entry and simulation
- Xilinx ISE synthesis and physical design tools
- Synplicity synthesis tools
- Synopsys synthesis and simulation tools
- Xilinx EDK for embedded platforms
- Cadence VLSI design tools
- Atmel WinAVR design tools
- Atmel Studio 4
- TinyOS

The hardware labs are supported by a Hardware Lab Manager, Bruce Hemingway, who orders parts and equipment as needed, organizes how the space is used, maintains parts and supplies used by the classes, and supervises a set of student lab assistants who provide support for the course labs. Bruce devotes half-time to lab management and half-time to teaching.

Software packages

Computing courses use a variety of Windows and Unix-based software tools, including:

- Java JDK
- Eclipse IDE
- CVS and SVN version control
- Squeak Smalltalk, Standard ML, Ruby, Perl, CGI, PHP
- bugzilla
- MySQL, SQL 2000, IIS, PostgreSQL
- VMWare
- Adobe Illustrator/Photoshop, Creative Suite 2
- AfterEffects 7.0. Deep Paint. RenderMan Artist Toolkit.
- Maya 6.0 & 7.0, AlfredForMaya, Premiere Pro 2.0

The department's computing infrastructure is supported and maintained by a staff of 15 computing professionals. Campus networking is supported by the University's Computing and Communications staff.

7 Institutional Support

The Department of Computer Science and Engineering has a strong foundation of funding provided by the State of Washington. The annual budget for the current fiscal year (FY07) including salaries and operations is \$8 million. The majority of our state funding supports personnel. An annual budget of \$4.8 million supports 41 tenure and tenure-track faculty and 3.5 instructors. State funding supports 6.5 advising staff, 9 technical staff and administrative staff including the Assistant to the Chair, Receptionist, Facilities Manager, Administrator and 3.5 fiscal staff. State funding is augmented from several sources including CSE's industrial affiliates program, corporate and individual gifts, and indirect cost returns (to the department) resulting from faculty awarded grants and contracts. These funds provide additional staff, a strong infrastructure and state-of-the-art facilities and equipment. Students do not have to manage machines themselves and can rely on a highly available and robust infrastructure for all their work and communications.

CSE receives funds through corporate and individual gifts and endowments. A major fundraising campaign provided \$40 million toward construction of a new building, and significantly increased the number of CSE's endowed professorships, chairs, fellowships and scholarships. The Innovation Endowment generates \$400,000 annually in discretionary funds. Over 60 companies contribute to CSE's industrial Affiliates Program annually. Over \$1 million in corporate equipment gifts were received in 2003 to support the additional computing labs in CSE's new building.

The UW Student Technology Fee Committee awarded \$57K in FY06 and \$89K in FY07 for improvements in instructional labs. The department also receives matching support in faculty recruiting from the College of Engineering and the Office of the Provost.

Computer Science and Engineering is in a strong fiscal position and resources are more than adequate to provide quality and excellence in our undergraduate program.

8 Computer Engineering Program Criteria

The curriculum for the computer engineering program provides depth and breadth across topics in computer science and computer engineering as described in detail in Section B.4 and Appendix I. The specific required mathematics, science and engineering knowledge is acquired as follows:

- Discrete mathematics is covered in depth in CSE 321 (Discrete Structures) and extended in other courses such as CSE 322 (Introduction to Formal Structures) and CSE 326 (Data Structures) and CSE 370 (Digital Design) which covers Boolean algebra in depth.
- Probability and Statistics is covered in STAT 390 or 391.
- Differential and integral calculus is covered in MATH 124, 125 and 126, the sequence on calculus and analytic geometry.
- Science background is covered in Physics 121 (Mechanics) and Physics 122 (Electromagnetism and Oscillatory Motion) and the natural science requirement.

Our computer engineering program focuses on the interface in computer systems between the hardware and software level. That is, computer engineers are faced with the design of systems that involve significant interaction between hardware and software and must have the knowledge required to deal with both sides of this interaction. Traditionally, this interface has been drawn close to the circuit level of hardware, where the computer engineer must have knowledge of analog and digital circuits, and understand computing systems comprised of these elemental

building blocks. Our hardware option focuses on this level of interface. Increasingly, however, engineers are confronted with a hardware/software interface at a higher and more abstract level that occurs in the design of operating systems, and networked and distributed systems. Our software option focuses on this higher level interface.

Students in both tracks learn the underlying fundamentals of circuits and electrical engineering in EE 215 (Introduction to Electrical Engineering) and the fundamentals of digital circuits in CSE 370 (Introduction to Digital Design). Both courses involve a substantial laboratory component and hands-on design. The computing hardware/software interface is then the central concept covered in CSE 378 (Machine Organization and Assembly Language), CSE 451 (Operating Systems) and CSE 461 (Introduction to Networks).

Students in the software track then take CSE 403 (Software Engineering) as well as electives from a set of courses that includes CSE 401 (Compiler Design), CSE 444 (Database Systems), CSE 466 (Embedded Systems) and CSE 471 (Computer Design), which exploit this hardware/software interface. Finally, in the software design capstone, students design and implement a substantial project that requires meeting a set of requirements while at the same time satisfying a set of constraints imposed by the real world.

Students in the hardware option extend their knowledge of low-level hardware concepts learned in EE 215 by taking EE 233 (Circuit Theory) and CSE 467 (Advanced Digital Design) where they get extensive hardware design experience in the lab. All hardware track students take CSE 466 (Embedded Systems) which exploits this low-level hardware/software interface. Students finish up with the CSE 477 Hardware Design Capstone where they design a complex system that involves the design of hardware and software, and the interface between them, to meet a set of real-world constraints. Many hardware option students also take CSE 468 (Introduction to VLSI) as well as senior EE courses.

In summary, students in the computer engineering program, whether in the software or hardware option learn the concepts required to analyze and design complex systems comprising hardware and software at the level appropriate to their specialization.

Appendix I

A. Tabular Data for Program

Table I-1. Basic level Curriculum

Table I-2. Course and Section Size Summary

Table I-3. Faculty Workload Summary

Table I-4. Faculty Analysis

Table I-5. Support Expenditures

Table I-1. Basic-Level Curriculum - Computer Engineering

Year; Semester or Quarter	Course (Department, Number, Title)	Category (Credit Hours)			
		Math & Basic Sciences	Engineering Topics Check if Contains <i>Significant Design</i> (✓)	General Education	Other
1 Yr/ 1 st Qtr	Math 124 Calc w/Anal Geometry	5	()		
	Natural Science Elective	5	()		
	English Composition		()	5	
1 Yr/ 2 nd Qtr			()		
	Math 125 Calc w/Anal Geometry	5	()		
	Phys 121/131 Mechanics	5	()		
	VLPA/I&S Elective		()	5	
1 Yr/ 3 rd Qtr			()		
	Math 126 Calc w/Anal Geometry	5	()		
	Physics 122/132 Electromagnetism	5	()		
	VLPA/I&S Elective		()	5	
2 Yr/ 1 st Qtr	Natural Science Elective	5	()		
	CSE 142 Computer Programming I		4 (✓)		
	VLPA/I&S Elective		()	5	
			()		
2 Yr/ 2 nd Qtr	Math 308 Linear Algebra	3	()		
	CSE 143 Computer Programming II		5 (✓)		
	VLPA/I&S Elective		()	4	
	VLPA/I&S Elective		()	4	
2 Yr/ 3 rd Qtr			()		
	Math/Stat 390 Prob & Statistics	4	()		
	EE 215 Fundamentals of EE		4 (✓)		
	TC 231 Intro. to Technical Writing		()	3	
	VLPA/I&S Elective		()	4	
3 Yr/ 1 st Qtr	CSE 303 Concepts&Tools for SW		3 ()		
	CSE 370 Intro to Digital Design		4 (✓)		
	EE 233 Circuit Theory (HW option)		5 (✓)		
	Free Elective		()		3-8
			()		
3 Yr/ 2 nd Qtr	CSE 378 Mach Org/Assembly Lang		4 (✓)		
	CSE 321 Discrete Structures	2	2 ()		
	VLPA/I&S Elective		()	5	
			()		
3 Yr/ 3 rd Qtr	CSE 341 Programming Languages		4 (✓)		
	CSE 322 Formal Models	1	2 ()		
	CSE 326 Data Structures		4 (✓)		
	Free Elective		()		4

Table I-1. Basic-Level Curriculum - Computer Engineering (continued)

Year; Semester or Quarter	Course (Department, Number, Title)	Category (Credit Hours)			
		Math & Basic Science	Engineering Topics <i>Check if Contains Significant Design (✓)</i>	General Education	Other
4 Yr/ 1 st Qtr Hardware Option	CSE 451 Operating Systems		4 (✓)		
	CSE 461 Networks		4 (✓)		
	CSE 467 Advanced Digital Design		4 (✓)		
	Free Elective		()		3
			()		
4 Yr/ 2 nd Qtr	CSE 466 SW for Embedded Systems		4 (✓)		
	Free Elective		()		3
	CompE Elective		3 ()		
	Free Elective		()		5
			()		
4 Yr/ 3 rd Qtr	CSE 477 Digital Systems Capstone		5 (✓)		
	TC 333 Adv. Tech Writing & Oral Presentation		()	4	
	CompE Elective		3 ()		
	Free Elective		()		3
4 Yr/ 1 st Qtr Software Option	CSE 403 Software Engineering		4 (✓)		
	CSE 451 Operating Systems		4 (✓)		
	CompE Elective		4 ()		
	Free Elective		()		3
			()		
4 Yr/ 2 nd Qtr	CSE 461 Networks		4 (✓)		
	Two of CSE 401 (Compilers), 421 (Algorithms), 444 (Databases), 466 (Embedded Systems), 471 (Computer Design)		6-8 (✓)		
	Free Elective		()		3-5
			()		
4 Yr/ 3 rd Qtr	CSE 4xx Software Systems Capstone		5 (✓)		
	TC 333 Adv. Tech Writing & Oral Presentation		()	4	
	CompE Elective		3 ()		
	Free Elective				3
TOTALS-ABET BASIC-LEVEL REQUIREMENTS		45	66-68 ¹	42	25-32
OVERALL TOTAL FOR DEGREE	180				
PERCENT OF TOTAL					
Totals must satisfy one set	Minimum quarter credit hours	45 hrs	67.5 hrs		
	Minimum percentage	25%	37.5 %		

¹ The shortfall in credits for the software track and our response is discussed in Section 4. The curriculum requirements that will be in effect starting next year are shown in the Table I-1a.

**Table I-2. Course and Section Size Summary
Computer Engineering**

Course No.	Title	No. of Sections offered in Current Year	Avg. Section Enrollment	Type of Class ¹			
				Lecture	Laboratory	Recitation	Other
100	Computer Fluency	3	59	60		40	
130	Pixels, Numbers and Programs	1	16	75		25	
142	Computer Programming I	7	211	75		25	
143	Computer Programming II	4	193	60		40	
190L	Current Topics -- OO Programming	1	89	75		25	
190M	Current Topics -- Web Programming	1	96	75	25		
303	Concepts&Tools for Software Dev.	3	56	100			
321	Discrete Structures	4	46	75		25	
322	Formal Models	3	53	100			
326	Data Structures	5	35	75		25	
341	Programming Languages	3	53	75		25	
370	Introduction to Digital Design	3	50	50	50		
373	Data Structures and Algorithms	3	50	100			
378	Machine Org. and Assembly Lang	3	49	75		25	
401	Introduction to Compiler Construction	2	32	100			
403	Software Engineering	3	33	75		25	
410	Computer Systems	1	24	100			
413	Prog. Langs and Implementation	1	10	100			
415	Introduction to Artificial Intelligence	1	28	100			
417	Algorithms and Comp. Complexity	1	14	100			
421	Introduction to Algorithms	2	46	100			
431	Introduction to Theory of Comp.	1	25	100			
444	Introduction to Database Systems	2	52	100			
450	Animation Production Seminar	1	104	100			
451	Introduction to Operating Systems	3	48	75		25	
454	Advanced Internet and Web Services	1	27	100			
455	Computer Vision	1	37	100			

(continued on next page)

Course No.	Title	No. of Sections offered in Current Year	Avg. Section Enrollment	Type of Class ¹			
				Lecture	Laboratory	Recitation	Other
456	Story Design for Computer Animation	1	20	60			40
457	Computer Graphics	2	48	100			
458	Computer Animation	1	21	60			40
459	Animation Production	1	19	60			40
460	Animation Capstone	1	19	60			40
461	Introduction to Networks	4	29	75		25	
466	Software for Embedded Systems	1	28	75	25		
467	Advanced Logic Design	1	27	75	25		
471	Computer Design and Org.	1	15	75		25	
472	Computational Linguistics	1	6	75		25	
473	Introduction to Artificial Intelligence	2	41	100			
477	Digital System Design	1	23	60	40		
481A	Operating Systems Capstone	1	11	60			40
481B	Tablet Computing Capstone	1	16	60			40
481D	Software Design Capstone (Games)	1	24	60			40
481E	Urban Sim Capstone	1	12	60			40
481F	Computational Biology Capstone	1	11	60			40
490C	Computational Biology	1	13	100			
490F	HCI	2	21	100			
490h	Software Design & Development	2	14	100			
490I	Neurobotics	1	26	50	50		
490K	Computer Security	1	39	100			

**Table I-3. Faculty Workload Summary
Computer Engineering**

Faculty Member (Name)	FT or PT (%)	Classes Taught (Course No./Credit Hrs.) Term and Year1	Total Activity Distribution ²		
			Teaching	Research	Other ³
Anderson, Richard	FT	Au06-421(3); Wi07-481(5);	33	50	17
Anderson, Tom	FT	Wi07-561(3); Sp07-552(3);	33	50	17
Bacon, Dave	FT	Wi07-326(4);	33	50	17
Balazinksa, Magda	FT	Au06-544(3); Wi07-303(3);	33	50	17
Beame, Paul	FT	Au06-321(4); Wi07-322(3); Sp07-431(3);	33	50	17
Bershad, Brian	FT	Sabbatical			100
Borning, Alan	FT	Au06-341(4); Sp07-481(5);	33	50	17
Borriello, Gaetano	FT	Au06-370(4); Wi07-466(4); Sp07-477(5);	33	50	17
Chambers, Craig	FT	Sabbatical			100
Curless, Brian	FT	Au06-457(4); Sp07-457(4);	33	50	17
Domingos, Pedro	FT	Au06-573(3); Sp07-P546(4);	33	50	17
Ebeling, Carl	FT	Au06-467(4); Wi07-567(3);	33	50	17
Eggers, Susan	FT	Au06-P548(4); Sp07-471(4);	33	50	17
Etzioni, Oren	FT	Au06-454(5); Sp07-574(3);	33	50	17
Fogarty, James	FT	Wi07-599(3); Sp07-510(3);	33	50	17
Fox, Dieter	FT	Sabbatical			100
Gribble, Steve	FT	Sp07-551(3);	17	25	58
Grossman, Dan	FT	Au06-505(3); Wi07-506(3); Sp07-303(3);	33	50	17
Guruswami, Venkat	FT	Au06-523(3); Wi07-531(3); Sp07-321(4);	33	50	17

Faculty Member (Name)	FT or PT (%)	Classes Taught (Course No./Credit Hrs.) Term and Year1	Total Activity Distribution ²		
			Teaching	Research	Other ³
Halevy, Alon	PT	Leave			100
Hemmingway, Bruce	PT	Wi07-370(4); Sp07-370(4);	33	50	17
Karlin, Anna	FT	Au06-521(3); Wi07-421(3)&P521(4);	33	50	17
Kautz, Henry	FT	Sabbatical	33	50	17
Kohno, Yoshi	FT	Au06-590(3); Wi07-P590(4); Sp07-490(4);	33	50	17
Krishnamurthy, Arvind	FT	Wi07-461(4);	33	50	17
Ladner, Richard	FT	Sabbatical	33	50	17
Landay, James	FT	Au06-490(3); Wi07-490(3); Sp07-599(3);	33	50	17
Lazowska, Ed	FT	Au06-P590(4); Wi07-451(4);	33	50	17
Lee, James	FT	Au06-321(4); Wi07-599(3); Sp07-525(3);	33	50	17
Levy, Hank	FT	Chair			100
Matsuoka, Yoki	FT	Wi07-490(4); Sp07-590(3);	33	50	17
Mones, Barbara	FT	Sabbatical			100
Notkin, David	FT	Sabbatical			100
Oskin, Mark	FT	Au06-378(4); Wi07-549(3); Sp07-378(4);	33	50	17
Perkins, Hal	FT	Au06-413(3)&401(3); Wi07-373(3) &	80		20
Popovic, Zoran	FT	Au06-P557(4); Wi07-557(3); Sp07-558(3);	33	50	17
Rao, Rajesh	FT	Au06-473(3); Sp07-528(3);	33	50	17
Reges, Stuart	FT	Au06-143(5)/599IT(3); Wi07-143(5)/341(4);	80		20
Ruzzo, Larry	FT	Au06-527(3); Wi07-417(3); Sp07-341(4);	33	50	17
Seitz, Steve	FT	Sabbatical			100
Shapiro, Linda	FT	Au06-EE562(3); Wi07-455(3); Sp07-576(3);	33	50	17
Snyder, Larry	FT	Au06-326(4); Wi07-401(3); Sp07-P524(4);	33	50	17
Stepp, Marty	FT	Au06-142(4)(3); Wi07-142(4)&403(4);	80		20
Suciu, Dan	FT	Au06-444(3); Wi07-P544(3); Sp07-444(3);	33	50	17
Tanimoto, Steven	FT	Au06-373(3); Wi07-415(3); Sp07-	33	50	17

Faculty Member (Name)	FT or PT (%)	Classes Taught (Course No./Credit Hrs.) Term and Year ¹	Total Activity Distribution ²		
			Teaching	Research	Other ³
Tompa, Martin	FT	Au06-322(3); Wi07-490(3); Sp07-481(5);	33	50	17
Weld, Dan	FT	Sabbatical			100
Wetherall, David	FT	Leave			100
Zahorjan, John	FT	Au06-461(4); Wi07-378(4); Sp07-481(5);	33	50	17

**Table I-4. Faculty Analysis
Computer Engineering**

Name	Rank	FT or PT	Highest Degree	Institution from which Highest Degree Earned & Year	Years of Experience			State in which Registered	Level of Activity (high, med, low, none)		
					Govt./ Industry Practice	Total Faculty	This Institution		Professional Society (Indicate Society)	Research	Consulting /Summer Work in Industry
Anderson, Richard	Professor	FT	PhD	Stanford, 1985	0	21	21		L	M	N
Anderson, Tom	Professor	FT	PhD	Washington, 1991	4	16	10		M	M	L
Baer, Jean-Loup	Professor Emeritus	FT	PhD	UCLA, 1968	0	42	38		H	H	N
Balazinska, Magda	Assist. Professor	FT	PhD	MIT, 2006	0	2	2		H	H	L
Beame, Paul	Professor	FT	PhD	Toronto, 1987	0	20	20		M	H	L
Bershad, Brian	Assoc. Professor	FT	PhD	Washington, 1990	3	17	10		H	H	L
Borning, Alan	Professor	FT	PhD	Stanford, 1979	3	27	27		M	H	L
Borriello, Gaetano	Professor	FT	PhD	Berkeley, 1988	3	20	20		L	H	M
Chambers, Craig	Professor	FT	PhD	Stanford, 1992	0	16	16		M	H	N
Curless, Brian	Assoc. Professor	FT	PhD	Stanford, 1997	0	9	9		L	H	L
Diorio, Chris	Assoc. Professor	FT	PhD	Caltech, 1997	7	9	9		M	H	M
Domingos, Pedro	Assoc. Professor	FT	PhD	UC-Irvine, 1997	0	10	8		L	H	L
Dooley, Emer	Lecturer	PT	PhD	Washington, 2000	8	7	7		L	L	N
Ebeling, Carl	Professor	FT	PhD	CMU, 1986	0	21	21		M	H	L
Eggers, Susan	Professor	FT	PhD	Berkeley, 1989	4	18	18		L	H	L
Etzioni, Oren	Professor	FT	PhD	CMU, 1990	0	16	16		M	M	M

Name	Rank	FT or PT	Highest Degree	Institution from which Highest Degree Earned & Year	Years of Experience			State in which Registered	Level of Activity (high, med, low, none)		
					Govt. / Industry Practice	Total Faculty	This Institution		Professional Society (Indicate Society)	Research	Consulting /Summer Work in Industry
Fox, Dieter	Assoc. Professor	FT	PhD	Bonn, 1998	0	9	7		L	H	L
Gribble, Steve	Assoc. Professor	FT	PhD	Berkeley, 2000	0	7	7		L	H	N
Grossman, Dan	Assist. Professor	FT	PhD	Cornell, 2003	0	4	4		L	H	L
Guruswami, Venkat	Assist. Professor	FT	PhD	MIT, 2001	0	5	5		L	H	M
Halevy, Alon	Professor	FT	PhD	Stanford, 1993	4	9	8		M	H	H
Hemingway, Bruce	Lecturer	PT	BA	Indiana U. South Bend 1973	32	7	5		L	L	L
Karlin, Anna	Professor	FT	PhD	Stanford, 1987	6	13	11		M	H	N
Ladner, Richard	Professor	FT	PhD	Berkeley, 1971	0	36	36		L	H	M
Landay, James	Assoc. Professor	PT	PhD	CMU, 1996	2	11	3		L	H	M
Lazowska, Ed	Professor	FT	PhD	Toronto, 1977	0	30	30		H	L	H
Lee, James	Assist. Professor	FT	PhD	Berkeley, 2005	0	1	1		L	H	N
Levy, Henry	Professor	FT	MS	Washington, 1981	8	24	24		H	H	L
Mones, Barbara	Senior Lecturer	FT	MFA	Rhode Island School of Design, 1979	3	21	8		H	L	M
Notkin, David	Professor	FT	PhD	CMU, 1984	0	23	23		H	H	M
Oskin, Mark	Assist. Professor	FT	PhD	UC Davis, 2001	0	7	7		H	H	M
Perkins, Hal	Senior Lecturer	FT	MS	Cornell, 1982	2	24	9		L	N	N
Popovic, Zoran	Assoc. Professor	FT	PhD	CMU, 1999	0	8	8		N	H	N
Rao, Rajesh	Assist. Professor	FT	PhD	Rochester, 1998	2	7	7		M	H	L
Reges, Stuart	Senior Lecturer	FT	MS	Stanford, 1982	0	25	3		M	L	L
Ruzzo, Larry	Professor	FT	PhD	Berkeley, 1978	0	30	30		N	H	N
Salesin, David	Professor	PT	PhD	Stanford, 1991	5	16	16		L	H	H

**Table I-5. Support Expenditures
Computer Engineering**

Fiscal Year	1	2	3	4
	FY 2005 (prior to previous year)	FY 2006 (previous year)	FY 2007* (current year)	FY 2008* (year of visit)
Expenditure Category				
Operations ¹ (not including staff)	\$819,498	\$957,761	\$1,101,931	\$987,000
Travel ²	\$120,027	\$150,136	\$169,354	\$177,822
Equipment ³	\$520,715	\$412,324	\$371,377	\$325,000
Institutional Funds	(\$206,254)	\$88,936	\$68,506	75,000
Grants and Gifts ⁴	\$726,969	\$323,388	\$302,871	\$250,000
Graduate Teaching Assistants	\$681,184	\$546,402	\$509,673	\$530,060
Part-time Assistance ⁵ (other than teaching)	\$259,526	\$340,878	\$363,119	\$374,013

All amounts are for the entire department. Computer Engineering students account for approximately 38% of our students.

*Projected FY07 and FY08 expenditures.

B. Course Syllabi

B.1 Required Core Courses

CSE142: Computer Programming I (4)

Catalog description

Basic programming-in-the-small abilities and concepts including procedural programming (methods, parameters, return values), basic control structures (sequence, if/else, for loop, while loop), file processing, arrays and an introduction to defining objects.

Prerequisites

none

Textbook(s) and/or other required material

Building Java Programs, Addison Wesley, Stuart Reges and Marty Stepp

Course objectives

Students will master basic procedural programming constructs. They will learn to write and debug small programs (50-100 lines) using a full range of procedural techniques using a variety of input sources (console, file) and a variety of output destinations (console, file, graphical). Students will learn the design principles that are relevant to this style of “programming in the small” including decomposition, information hiding, elimination of redundancy, detailed documentation, and use of parameters and return values to create flexible components. Students will be exposed to the major design issues of object oriented programming including inheritance.

Topics covered

- * Basic concepts of computer structure and program execution
- * Variables, types, expressions, and assignment
- * Input/output: console, file, graphical
- * Conditional execution (if/else)
- * Iteration (for, while)
- * Defining methods: parameters, return values
- * Arrays (one-dimensional)
- * User-defined classes

- * Use of standard library objects: strings, graphics
- * Procedural decomposition of problems
- * Programming style: eliminating redundancy, localizing variables, class constants, commenting, use of parameters and return values to increase flexibility, appropriate choice of control structure (e.g., sequential if versus if/else)

Course structure

3 50-minute lectures per week
1 50-minute quiz section per week
Weekly projects to be completed outside of class

CSE143: Computer Programming II (5)

Catalog description

Continuation of 142. Concepts of data abstraction and encapsulation including stacks, queues, linked lists, binary trees, recursion, instruction to complexity and use of predefined collection classes.

Prerequisites

CSE 142.

Textbook(s) and/or other required material

Building Java Programs, Addison Wesley, Stuart Reges and Marty Stepp
Lecture notes on advanced topics

Course objectives

Students learn about data abstraction and the basic design principles of object oriented programming. Students will become familiar with standard data abstractions (lists, maps, sets, stacks, queues) as well as a variety of implementation techniques (arrays, linked lists, binary trees). Students will learn to program using recursion and recursive backtracking. Students will learn the object-oriented constructs that support code reuse (encapsulation, interfaces, inheritance, abstract classes) as well as learning how to make use of off-the-shelf components from libraries like the Java Collections Framework.

Topics covered

Abstract data types: stacks, queues, lists, maps, sets
Implementing linked structures (linked lists, binary trees)
Recursion and recursive backtracking
Using off-the-shelf components (e.g., Java Collections Framework)
Use of inheritance for additive change and factoring out common code into abstract classes
Class design: encapsulation, documentation, throwing exceptions, appropriate choice of fields
Thorough testing and debugging
Time and space complexity
Efficient sorting and searching algorithms (binary search, mergesort)
Iterator use and implementation

Course structure

3 50-minute lectures per week
2 50-minute quiz sections per week
Weekly projects to be completed outside of class

CSE303: Concepts and Tools for Software Development (3)

Catalog description

Introduction to key concepts and tools in the development of software not introduced in the introductory programming courses. Includes programming with explicit memory management and layout (e.g. C or C++), techniques for group software development, modern design, implementation, and testing patterns and strategies, and societal impact.

Prerequisites

CSE 143.

Textbook(s) and/or other required material

Linux Pocket Guide by Daniel J. Barrett, O'Reilly, 2004.

Programming in C (3rd Edition) by Stephen G. Kochan, Sams Publishing, 2005.

Course objectives

A primary goal of this course is to provide students with abilities, skills, and concepts that will make them more efficient software developers. More specifically, the class teaches students:

- * The skills necessary to automate common computing tasks including file-manipulation and string-processing.
- * C and C++-level programming with emphasize on memory management and pointers.
- * Common programming tools including debuggers, profilers, linkers, and version-control systems.

Because programming systems are themselves interesting and often quite beautiful (or at least elegant), the class also teaches students how to develop a sense of what distinguishes well designed from poorly designed systems. The class teaches software-engineering practices regarding specification and testing.

Because it is important for people engaging in professional activities to think about how their work relates to the larger social context, the second main goal of this class is to make students sensitive to the societal and ethical implications of computing. A portion of this course is set aside for discussions on this topic. The class teaches students how to critically evaluate several societal/ethical implications of computing and develop the intellectual maturity to evaluate new issues as they arise.

Topics covered

Topic 1: Societal and ethical implications of software

- * Professional and ethical responsibility
- * Impact of computer engineering solutions in a global, economic, environmental, and societal context
- * Contemporary societal and ethical issues

Topic 2: Expediting and automating tasks:

- * The Linux operating systems and the shell
- * Manipulating files and strings through the command line
- * Utilities (e.g., find, awk, sed, grep)
- * Shell scripts

Topic 3: Programming in C

- * Emphasize on memory management (stack and heap) and pointers
- * Brief introduction to C++

Topic 4: Software development tools

- * Build tools
- * Debuggers
- * Version control systems
- * Profilers

Topic 5: Basic software engineering concepts

- * Specifications, interfaces, and testing
- * Multiperson programming
- * Security and defensive programming

Course structure

Class: 3 days per week, 1 hour

Assignments: approximately 6 assignments spread throughout the quarter

Essays: three short essays on societal and ethical implications of computing

Midterm and final exam

ABET Outcomes Assessed

- (f) an understanding of professional and ethical responsibility
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (j) knowledge of contemporary issues

Additional ABET Outcomes Covered

- (g) an ability to communicate effectively
 - (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice
-

CSE321: Discrete Structures (4)

Catalog description

Fundamentals of set theory, graph theory, enumeration, and algebraic structures, with applications in computing.

Prerequisites

CSE 143; either MATH 126, MATH 129, or MATH 136.

Textbook(s) and/or other required material

Discrete Mathematics and Its Applications, Rosen (6th Edition)

Course objectives

Provide students with the definitions and basic tools for reasoning about discrete mathematical objects useful for computer science and engineering.

Topics covered

- * propositional logic and predicate logic
- * set theory, functions
- * the integers: divisibility, modular arithmetic, primality, etc.
- * methods of proof and formal reasoning
- * mathematical induction
- * recursive definitions
- * counting
- * discrete probability theory
- * binary relations
- * graph theory and graph algorithms
- * applications, including RSA

Course structure

Meetings: 3 1-hour lectures per week, 1 recitation section hour per week

Assignments: weekly written assignments

Midterm plus final

ABET Outcomes Assessed

- (l) knowledge of probability and statistics
- (m) knowledge of discrete mathematics

Additional ABET Outcomes Covered

- (a) an ability to apply knowledge of mathematics, science, and engineering
-

CSE322: Introduction to Formal Models in Computer Science (3)

Catalog description

Finite automata and regular expressions; context-free grammars and pushdown automata; nondeterminism; Turing machines and the halting problem. Emphasis on understanding models and their applications and on rigorous use of basic techniques of analysis. Induction proofs, simulation, diagonalization, and reduction arguments.

Prerequisites

CSE 321.

Textbook(s) and/or other required material

Michael Sipser, Introduction to the Theory of Computation, Second Edition.
Thompson Course Technology, 2006

Course objectives

Teach students the basics of three formal models of computation: finite automata, context-free languages, and Turing machines with special emphasis on methods for manipulating and reasoning about them.

Give students an idea of the value and applicability of the formal models.

Give sufficient grounding in automata and formal languages to prepare for parsing in the compilers class.

Languages, finite automata, regular expressions, context-free grammars, and other automata such as pushdown store machines and Turing machines. Introduction to proofs about computation models and theoretical concepts such as nondeterminism.

Topics covered

strings and languages:
concatenation, powers, and reversal of strings

regular languages:
regular expressions
finite automata
DFAs
NFAs
cross product construction
subset construction
Kleene's theorem
closure properties - union, concatenation, star, intersection, complement
decision problems - emptiness, finiteness, equivalence
nonregular languages
optional:
Myhill-Nerode Theorem
minimization of DFAs
pattern matching

context-free languages:
context-free grammars
examples from real programming languages such as C++
derivation trees
leftmost derivations
ambiguous grammars
closure properties - union, concatenation, star
Chomsky normal form
non-context-free languages

non-closure properties: intersection, complement
parsing methods
top-down and bottom-up parsing
efficient deterministic parsing (Cocke-Kasami-Younger algorithm)
pushdown automata
equivalence of PDAs and context-free grammars
deterministic PDAs
optional:
decision problems
inherently ambiguous context-free grammars

Turing machines and undecidability:
countability and uncountability (Cantor's proof)
universal Turing machine
halting problem is undecidable

Course structure

3 lecture hours per week. Weekly written assignments. Midterm and final exam.

ABET Outcomes Assessed

(m) knowledge of discrete mathematics

Additional ABET Outcomes Covered

- (a) an ability to apply knowledge of mathematics, science, and engineering
 - (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
 - (e) an ability to identify, formulate, and solve computer engineering problems
-

CSE326: Data Structures (4)

Catalog description

Data types, abstract data types, and data structures. Efficiency of algorithms. Sequential and linked implementation of lists. Binary tree representations and traversals. Searching: dictionaries, priority queues, hashing. Directed graphs, depth-first algorithms. Garbage collection. Dynamic storage allocation. Internal and external sorting. No credit to students who have completed CSE 373, CSE 374, or E E 374.

Prerequisites

CSE 321.

Textbook(s) and/or other required material

Data Structures and Algorithm Analysis in Java 2nd Ed., Mark Allen Weiss, Addison Wesley: 2007, ISBN: 0-321-37013-9

Course objectives

The objective of this class is to study the fundamental abstract data types and their implementations as data structures, together with efficient algorithms employing these data structures and their asymptotic analyses.

Topics covered

1. Introduction: Data Structures, Abstraction, Recursion
2. Algorithm Analysis
3. Dictionary Abstract Data Type
4. Search Trees: Binary, AVL, 2-3, Splay, B-trees
5. Hashing
6. Priority Queues: Heaps
7. Disjoint Sets: Weighted Union and Find with Compression
8. Sorting: Heapsort, Mergesort, Quicksort, Decision Trees
9. Graph Algorithms and Graph Representations

Course structure

3 hours per week of lecture

1 hour per week of discussion section

ABET Outcomes Assessed

(m) knowledge of discrete mathematics

Additional ABET Outcomes Covered

- (a) an ability to apply knowledge of mathematics, science, and engineering
 - (e) an ability to identify, formulate, and solve computer engineering problems
 - (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice
-

CSE341: Programming Languages (4)

Catalog description

Basic concepts of programming languages, including abstraction mechanisms, types, and scoping.
Detailed study of several different programming paradigms, such as functional, object-oriented, and logic programming. No credit if CSE 413 has been taken.

Prerequisites

CSE 143.

Textbook(s) and/or other required material

Varies depending on instructor, but typically one or two of:

Simon Thompson, Miranda: The Craft of Functional Programming
Jeff Ullman, Elements of ML Programming
Mark Guzdial, Squeak: Object-Oriented Programming with Multimedia Applications
Kent Dybvig, The Scheme Programming Language

(In addition, significant on-line materials, tutorials, and resources are provided.)

Course objectives

- * To understand fundamental programming-language concepts.
- * To become fluent in non-imperative programming paradigms.
- * To become able to learn new programming languages efficiently.

Topics covered

The following topics are always covered:

- * Functional programming (avoiding mutation; exploiting recursion and higher-order functions; closures; anonymous functions)
- * algebraic datatypes and pattern-matching
- * essential object-oriented programming (late-binding / dynamic dispatch, subtyping vs. subclassing)
- * language support for abstraction, such as modules, abstract types, and dynamic type-creation
- * syntax vs. semantics
- * static vs. dynamic typing
- * parametric polymorphism / generics
- * object-oriented extensibility vs. functional extensibility

In any particular offering, most of the following are covered:

- * tail recursion
- * currying
- * equality vs. identity
- * macros
- * type inference
- * lazy evaluation and related idioms such as streams and memoization
- * code-as-data concepts, such as reflection and eval/apply
- * lexical vs. dynamic scope
- * subtyping issues such as structural vs. named subtyping
- and sumption vs. coercion
- * object-oriented concepts such as multiple inheritance, multimethods, and metaclasses
- * bounded parametric polymorphism
- * forms of parameter passing
- * subtype polymorphism and bounded polymorphism
- * logic programming

- * garbage collection

A small number of these topics might also be covered:

- * language-design principles
- * history of programming languages
- * programming environments
- * debugging support
- * compilers vs. interpreters
- * continuations
- * continuation-passing style
- * coroutines
- * iterators
- * language support for concurrency

Course structure

3 lectures per week

1 section per week

ABET Outcomes Assessed

(k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

Additional ABET Outcomes Covered

(e) an ability to identify, formulate, and solve computer engineering problems

CSE370: Introduction to Digital Design (4)

Catalog description

Introductory course in digital logic and its specification and simulation. Boolean algebra, combinational circuits including arithmetic circuits and regular structures, sequential circuits including finite-state-machines, use of programmable logic devices. Simulation and high-level specification techniques are emphasized.

Prerequisites

none

Textbook(s) and/or other required material

Contemporary Logic Design (2nd Edition)

Randy H. Katz, Gaetano Borriello

2004, Prentice-Hall/Pearson

ISBN: 0201308576

Course objectives

Understanding of digital logic at the gate and switch level including both combinational and sequential logic elements.

Understanding of the clocking methodologies necessary to manage the flow of information and preservation of circuit state.

An appreciation for the specification methods used in designing digital logic and the basics of the compilation process that transforms these specifications into logic networks.

Facility with computer-aided design tools for digital logic design with programmable logic devices as the implementation technology.

To begin to appreciate the difference between hardware and software implementations of a function and the advantages and disadvantages of each.

Topics covered

Introduction: 1 lecture

Combinational Logic: 4 lectures

Working with Combinational Logic: 3 lectures

Combination Logic Examples: 4 lectures

Sequential Logic: 3 lectures:

Implementation Technologies: 3 lectures

Hardware Description Languages: 2 lectures

Finite State Machines: 4 lectures

Sequential Logic Examples: 3 lectures

Computer Organization: 3 lectures

Course structure

Meetings: 3 1-hour lectures per week, 1 laboratory section (3 hours) per week

Assignments: weekly written assignments with a larger project as the last assignment

Laboratory: use of computer-aided design tools for aspects of almost every assignment

ABET Outcomes Assessed

(a) an ability to apply knowledge of mathematics, science, and engineering

Additional ABET Outcomes Covered

(c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(e) an ability to identify, formulate, and solve computer engineering problems

(k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

CSE378: Machine Organization and Assembly Language (4)

Catalog description

Differences and similarities in machine organization; central processors; fundamentals of machine language and addressing; assembly language programming, including macros; operating system interfaces. No credit to students who have completed 410.

Prerequisites

CSE 143; CSE 370.

Textbook(s) and/or other required material

D. Patterson and J. Hennessy:

Computer Organization & Design: The Hardware/Software Interface
(3rd Edition, 1998)

Course objectives

Fundamentals of instruction set design. CPU implementation, pipelining. Memory hierarchy. Assembly language programming. Compiler and operating system interfaces.

The purpose of this course is to give students a basic understanding of computer architecture and organization, assembly language programming, and the hardware/software interface.

Topics covered

machine organization, instruction sets, addressing modes, instruction encoding, subroutine linkages, assessing performance, datapath and control, pipelining, caches, address translation, I/O, exception and protection mechanisms.

Course structure

3 lectures/week

1 discussion section/week

ABET Outcomes Assessed

(b) an ability to design and conduct experiments, as well as to analyze and interpret data

(c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

Additional ABET Outcomes Covered

(a) an ability to apply knowledge of mathematics, science, and engineering

(e) an ability to identify, formulate, and solve computer engineering problems

(k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

CSE451: Introduction to Operating Systems (4)

Catalog description

Principles of operating systems. Process management, memory management, auxiliary storage management, resource allocation. No credit to students who have completed 410 or E E 474.

Prerequisites

CSE 326; CSE 378.

Textbook(s) and/or other required material

Silberschatz and Galvin, Operating Systems Concepts

Course objectives

Give students a working knowledge of operating systems principles, design issues, algorithms and data structures.

Build programming experience through a sequence of targeted OS projects.

Topics covered

operating system structure, processes, threads, synchronization, scheduling, deadlock, virtual memory, secondary storage management, distributed systems, file systems, security

Course structure

Class: 3 days per week, 1 hour

Sections: 1 day per week, 1 hour

Labs: as needed for project and assignments

ABET Outcomes Assessed

(b) an ability to design and conduct experiments, as well as to analyze and interpret data

(e) an ability to identify, formulate, and solve computer engineering problems

Additional ABET Outcomes Covered

(a) an ability to apply knowledge of mathematics, science, and engineering

(c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

CSE461: Introduction to Computer-Communication Networks (4)

Catalog description

Computer network architectures, protocol layers, network programming. Transmission media, encoding systems, switching, multiple access arbitration. Network routing, congestion control, flow control. Transport protocols, real-time, multicast, network security.

Prerequisites

CSE 143; either MATH 390/STAT 390, STAT 391, IND E 315, or CSE 321.

Textbook(s) and/or other required material

"Computer Networks: A Systems Approach", 3rd edition, Larry Peterson and Bruce Davie

Course objectives

To provide students with an understanding of how to construct large-scale computer networks. This includes an appreciation of the fundamental problems that arise in building networks, the design principles that are of proven value, and the common implementation technologies that are in use today.

Topics covered

This course introduces the basics of networking, ranging from transmitting bits over wires to the Web and distributed computing. We focus on the inter-networking issues in-between these two extremes. We will cover protocol layering in general and the following topics in particular: framing, error correction, packet and circuit switching, multi-access protocols (Ethernet), queuing, addressing and forwarding (IP), distance vector and link state routing, reliable transport, congestion control (TCP), and security.

Course structure

Class: 3 days per week, 1 hour

Sections: 1 day per week, 1 hour

Labs: as needed for project and assignments

ABET Outcomes Assessed

(c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(e) an ability to identify, formulate, and solve computer engineering problems

Additional ABET Outcomes Covered

(a) an ability to apply knowledge of mathematics, science, and engineering

(b) an ability to design and conduct experiments, as well as to analyze and interpret data

(k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

B.2 Specified Courses in Hardware and Software Tracks

CSE401: Introduction to Compiler Construction (3)

Catalog description

Fundamentals of compilers and interpreters; symbol tables; lexical analysis, syntax analysis, semantic analysis, code generation, and optimizations for general purpose programming languages. No credit to students who have taken 413.

Prerequisites

CSE 322; CSE 326; CSE 341; CSE 378.

Textbook(s) and/or other required material

Cooper and Torczon, Engineering a Compiler, Morgan-Coffman

Course objectives

Learn principles and practice of language implementations.

Understand tradeoffs between run-time and compile-time processing.

Understand tradeoffs between language features, run-time efficiency, and implementation difficulty.

Gain experience working with large systems software, object-oriented design, and Java.

Topics covered

Organization of Compilers and Interpreters

Lexical Analysis

Syntactic Analysis

Semantic Analysis

Interpretation

Run-Time Storage Layout

Code Generation

Optimization

Course structure

Lecture, 3 hours per week.

Laboratory, as needed to complete course project.

ABET Outcomes Assessed

(a) an ability to apply knowledge of mathematics, science, and engineering

(k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

Additional ABET Outcomes Covered

(c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(e) an ability to identify, formulate, and solve computer engineering problems

CSE403: Software Engineering (4)

Catalog description

Fundamentals of software engineering using a group project as the basic vehicle. Topics covered include the software crisis, managing complexity, requirements specification, architectural and detailed design, testing and analysis, software process, and tools and environments.

Prerequisites

CSE 326; CSE 341; recommended: project experience in an academic or work setting.

Textbook(s) and/or other required material

SOFTWARE PROJECT SURVIVAL GUIDE

S. McConnell

ISBN-10: 1-57231-621-7

THE PRAGMATIC PROGRAMMER: FROM JOURNEYMAN TO MASTER

A. Hunt, D. Thomas

ISBN 978-0201616224

Assorted papers from the literature

Course objectives

A central objective of the course is to have students develop a deep understanding of the distinctions between software engineering and programming. In addition, the students understand the software lifecycle, increase their knowledge of classic and modern software engineering techniques, and develop concrete experience in turning ill-formed concepts into products working with a team.

Topics covered

Topics covered include the software crisis, managing complexity, requirements specification, architectural and detailed design, testing and analysis, software process, and tools and environments.

Course structure

There are three lectures each week (50 minutes each) and one section with the teaching assistant. The lectures cover general software engineering material, material pertinent specifically to the quarter-long project, and usually includes some guest lectures from local industry. The sections are often used for reviews of the projects and for presentation by the TA of material on specific tools and technologies needed to develop the project.

ABET Outcomes Assessed

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

Additional ABET Outcomes Covered

- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
 - (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
 - (d) an ability to function on multi-disciplinary teams
 - (e) an ability to identify, formulate, and solve computer engineering problems
 - (j) knowledge of contemporary issues
-

CSE421: Introduction to Algorithms (3)

Catalog description

Techniques for design of efficient algorithms. Methods for showing lower bounds on computational complexity. Particular algorithms for sorting, searching, set manipulation, arithmetic, graph problems, pattern matching.

Prerequisites

CSE 322; CSE 326.

Textbook(s) and/or other required material

Kleinberg and Tardos, Algorithm Design.
Addison Wesley

Course objectives

Learn basic techniques for design and analysis of algorithms, including correctness proofs. Learn a number of important basic algorithms. Learn how to prove that problems are NP-complete.

Topics covered

Main Techniques:

Design: Induction, Graph search, Divide and Conquer, Greedy, Dynamic Programming, Network Flow

Analysis: Asymptotic Analysis, Recurrences.

Intractability: Reduction.

Typical Algorithm coverage: depth- and breadth-first search, bi- and/or strongly connected components, shortest paths, min spanning trees, transitive closure, flows and matchings, Strassen's method, FFT, knapsack, edit distance/string matching, scheduling.

Intractability: Reduction. P, NP, verification/certificates/witnesses, nondeterminism, reduction, completeness. Example problems: SAT, 3-SAT, clique, vertex cover, 0-1 knapsack, partition, coloring.

Course structure

3 50 minute lectures per week

or 2 80 minute lectures per week.

ABET Outcomes Assessed

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (e) an ability to identify, formulate, and solve computer engineering problems
- (m) knowledge of discrete mathematics

Additional ABET Outcomes Covered

CSE444: Introduction to Database Systems (3)

Catalog description

Fundamental concepts, system organization, and implementation of database systems. Relational, hierarchical, and network data models; file organizations and data structures; query languages; query optimization; database design; concurrency control; security; issues involving distributed database systems.

Prerequisites

CSE 326.

Textbook(s) and/or other required material

Hector Garcia-Molina, Jeffrey Ullman, Jennifer Widom. Database Systems: The Complete Book, Prentice-Hall, 2002.

Course objectives

Databases are at the heart of modern commercial application development. Their use extends beyond this to many applications and environments where large amounts of data must be stored for efficient update and retrieval. The purpose of this course is to provide an introduction to the design and use of database systems, as well as an appreciation of the key issues in building such systems.

We begin by covering the relational model and the SQL language. We then study methods for database design, covering the entity relationship model. Next, we discuss XML as a data model, and present languages for querying it. We see how XML is used for sharing data among different applications in a distributed environment. We then inspect the architecture of a database system, and discuss efficient storage of data, execution of queries and query optimization. Finally, we touch on some advanced topics in database systems.

Topics covered

data models, conceptual design, query languages, system components, data storage, query optimization and transaction processing.

Course structure

3 lectures a week

ABET Outcomes Assessed

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

Additional ABET Outcomes Covered

- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
 - (e) an ability to identify, formulate, and solve computer engineering problems
 - (j) knowledge of contemporary issues
-

CSE466: Software for Embedded Systems (4)

Catalog description

Software issues in the design of embedded systems. Microcontroller architectures and peripherals, embedded operating systems and device drivers, compilers and debuggers, timer and interrupt systems, interfacing of devices, communications and networking. Emphasis on practical application of development platforms.

Prerequisites

CSE 326; CSE 370; CSE 378.

Textbook(s) and/or other required material

Extensive use of technical datasheets and on-line documentation of devices and tools.

Course objectives

Understanding of basic microcontroller architecture and the motivation for their special features that distinguish them from microprocessors.

Understanding of interfacing techniques for connecting microcontrollers to a variety of sensors and actuators (both digital and analog).

An appreciation for the different considerations and constraints that software developers for embedded systems must deal with.

Appreciation of power management methods.

Understanding of basic communication protocols both wired and wireless.

Facility with a complete set of tools for embedded systems programming and debugging.

Experience with implementing several embedded systems with particular focus on their interaction between multiple devices and between devices and general-purpose computers.

Topics covered

Embedded Systems: applications and platforms

Microcontrollers: architectures, peripheral units, memory systems

Interrupts: interrupt architectures, device drivers, timers

Interfacing: basics of interfacing including GPIO, D/A, A/D, serial ports

Communication Protocols: serial and parallel communication methods

Introduction to Sensor Networks: basics of sensor networks including networking protocols

Sensor Node Software: embedded and real-time OS issues, modularity

Radio Protocols: common radio protocols and emerging standards

Project Design: developing the specification of the final project (flock, soccer, etc.)

TinyOS Case Study: bottom-up design of an embedded operating system

Industry Guest Lectures: lectures from local companies developing mobile and embedded systems

Special Topics: on-going embedded systems research projects at UW

Course structure

3 hours lecture

3 hours lab

ABET Outcomes Assessed

(a) an ability to apply knowledge of mathematics, science, and engineering

(c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

Additional ABET Outcomes Covered

(b) an ability to design and conduct experiments, as well as to analyze and interpret data

(e) an ability to identify, formulate, and solve computer engineering problems

CSE467: Advanced Digital Design (4)

Catalog description

Advanced techniques in the design of digital systems. Hardware description languages, combinational and sequential logic synthesis and optimization methods, partitioning, mapping to regular structures. Emphasis on reconfigurable logic as an implementation medium. Memory system design.

Digital communication including serial/parallel and synchronous/asynchronous methods.

Prerequisites

CSE 326; CSE 370.

Textbook(s) and/or other required material

No required text

Reference: Contemporary Logic Design (2nd Edition) Randy H. Katz, Gaetano Borriello 2004, Prentice-Hall/Pearson ISBN: 0201308576.

Additional material supplements lectures.

Course objectives

1. To learn how to design digital systems, from specification and simulation to construction and debugging.
2. To learn techniques and tools for programmable logic design
3. To learn how to use modern laboratory test equipment, including logic analyzers and oscilloscopes.
4. To understand the limitations and difficulties in modern digital design, including wiring constraints, high-speed, etc.
5. To design, construct, test, and debug a moderate-scale digital circuit.

Topics covered

Overview of digital technology

Logic families

TTL/CMOS

Reading and understanding data books

Interfaces

Standard components

Programmable devices

PROMs

PALs and PLDs

FPGAs

Electrical realities

Resistance, capacitance and inductance

Wire delays and time constants

Fanout and loading

Decoupling and signal integrity

Power dissipation and drops

Ringings, reflections, and terminations

Clock distribution

Computer-aided design

Hardware description languages (HDLs, esp. Verilog)

Logic compilation

Two-level and multi-level logic synthesis

Technology-independent optimization

Technology mapping

Sequential-logic synthesis

Tools for mapping to PLDs and FPGAs

Laboratory

Logic analyzer and oscilloscope basics

Timing, state, capture, bandwidth

Glitches and transient events

Debugging techniques

System-level components

Static, dynamic, and nonvolatile memories
RAM, ROM, PROM, EPROM, EEPROM
Memory controllers and timing
Digital communication
Serial and parallel protocols
Synchronous vs. asynchronous data communication
Busses
Arbitration

Course structure

Three 50-minute lectures per week
One 3-hour lab session per week
Assignments and a final project

ABET Outcomes Assessed

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

Additional ABET Outcomes Covered

- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
 - (e) an ability to identify, formulate, and solve computer engineering problems
-

CSE471: Computer Design and Organization (4)

Catalog description

CPU instruction addressing models, CPU structure and functions, computer arithmetic and logic unit, register transfer level design, hardware and microprogram control, memory hierarchy design and organization, I/O and system components interconnection. Laboratory project involves design and simulation of an instruction set processor.

Prerequisites

CSE 370; CSE 378.

Textbook(s) and/or other required material

J.Hennessy and D.Patterson "Computer Organization and Design: The Hardware/Software Interface", 4th ed. Morgan-Kaufman, 2007

Course objectives

Teach the design and architecture of major components of the structure of the central processing unit and memory hierarchy of modern microprocessor systems.

Use a cycle by cycle simulator to illustrate logic complexities.

Topics covered

- Pipelining
- Branch prediction
- Exceptions
- Examples of CPU's in modern microprocessors
- Exploiting Instruction Level Parallelism
- Scoreboard and Tomasulo's algorithm
- Superscalars
- Caches and cache assists
- Hardware assists for paging systems
- TLB's
- Symmetric MultiProcessors
- Cache coherence
- Synchronization

Course structure

2 lectures a week (1:20mns each)

ABET Outcomes Assessed

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

Additional ABET Outcomes Covered

- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve computer engineering problems

B.3 Hardware and Software Capstone Courses

CSE454: Advanced Internet and Web Services (5)

Catalog description

Design of Internet search engines, including spider architecture, inverted indices, frequency rankings, latent semantic indexing, hyperlink analysis, and refinement interfaces. Construction of scalable and secure web services. Datamining Webserver logs to provide personalized and user-targeted services. Large project.

Prerequisites

CSE 326.

Textbook(s) and/or other required material

Selected papers available on the Internet.

Course objectives

Understand the intellectual foundations of Web Search and Internet technologies including information retrieval, data mining, and cryptography. Be able to understand, build, and debug a Web Search Engine.

Topics covered

September 28, 2006: Introduction, History & Networking Summary

October 3, 2006: HTTP and Servers

October 5, 2006: Page Generation

October 10, 2006: Web Service Operations

October 17, 2006: Inverted Indexes

October 19, 2006: IR

October 24, 2006: PageRank

October 26, 2006: Relevance Feedback

October 31, 2006: Machine Learning, and pan-lingual material

November 7, 2006: Text categorization

November 9, 2006: Text categorization

November 14, 2006: Clustering .

November 16, 2006: Information Extraction

Course structure

The course has discussion/lecture sessions bi-weekly and a course project that culminates in a report and student oral presentations. More detail on the project is below.

The basic idea is to build a Google-style search engine which is great at finding pages. We will focus especially on how to rank large numbers of search results.

For this class, you'll have access to the usual undergrad machines (such as barb) as well as two special compute nodes: kiska and umnak. If you have some experiments to run that are especially compute-intensive, try to run them there.

Your early projects won't consume much disk space, but you will be eating quite a lot of it by the end of the class. Don't worry: we will arrange to create student workspace areas before you need the space.

Project 1, Text Ranker, is due at 12:00noon, Tuesday, October 24, 2006. (For fun, see the support code src [here](#).)

Project 2, Web Ranker, is due at 12:00noon, Tuesday, November 7, 2006. (For fun, see the support code src [here](#).)

Project 3, Final Project, has multiple due dates.

ABET Outcomes Assessed

(b) an ability to design and conduct experiments, as well as to analyze and interpret data

(c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to, engage in life-long learning
- (j) knowledge of contemporary issues

Additional ABET Outcomes Covered

- (a) an ability to apply knowledge of mathematics, science, and engineering
 - (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice
-

CSE477: Digital System Design (5)

Catalog description

Capstone design experience. Prototype a substantial project mixing hardware, software, and communication components. Focuses on use of embedded processors and programmable logic in digital system design, case studies, and emerging components and platforms. Provides a complete experience in embedded system design and management.

Prerequisites

CSE 451; CSE 466; CSE 467.

Textbook(s) and/or other required material

Required material, including datasheets and documentation of components and software, is handed out in class.

Course objectives

To serve as a capstone design course to tie together the computer engineering curriculum via the design of a complete embedded system involving multiple communicating components.

To gain appreciation for the interaction between hardware and software in embedded system design.

Familiarity with basic inter-component communication methods.

To experience the development of a complete product from design to implementation and debugging.

To present design goals and decisions as well as implementation results in both verbal presentation and written documentation.

To work toward a common goal in a team environment.

Topics covered

Introduction: Embedded Systems; Product design; Product development process

Team-based Design: Project definition; Decomposition into hardware and software components;

Evaluation of needs; Assessment of team members' skills; Experimental design to resolve unknowns;

Design reuse

Project Evolution: Design for maintainability; Design for upgradability; Modularity in Design

Research Directions: Architecture innovations; Operating systems; Networking; Application domains;

User interfaces

Case Studies: Past capstone design projects; Industry case studies; Research projects in department

Course structure

The class meets three hours a week for lectures. Lecture time is spent developing a theme for the projects and understanding prototyping and project management processes. Students give presentations on their projects for peer design reviews twice during the quarter. Lab time (as well as additional time) is spent designing, developing, and debugging project elements. Each group meets weekly with the instructor or TAs to develop and assess weekly milestones leading to completion of the project and a successful demonstration at the end of the quarter.

ABET Outcomes Assessed

(b) an ability to design and conduct experiments, as well as to analyze and interpret data

(c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(d) an ability to function on multi-disciplinary teams

(e) an ability to identify, formulate, and solve computer engineering problems

(f) an understanding of professional and ethical responsibility

(g) an ability to communicate effectively

(h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context

(i) a recognition of the need for, and an ability to, engage in life-long learning

(j) knowledge of contemporary issues

Additional ABET Outcomes Covered

(a) an ability to apply knowledge of mathematics, science, and engineering

(k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

CSE481A: Operating Systems Capstone (5)

Catalog description

Students work in substantial teams to design, implement, and release a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product. Teams are expected to develop a work plan, and to track and document their progress against it.

Prerequisites

Senior standing in CSE or permission of the instructor.

Specific description

Students work in substantial teams to design, implement, and release a software project involving multiple areas of the CSE curriculum. A lab containing the sources for a commercial operating system serves as the framework for the class.

Textbook(s) and/or other required material

“Microsoft Windows Internals Fourth Edition” by Mark E. Russinovich & David A. Solomon. Highly recommended.

Course objectives

Build on the fundamental computer operating system knowledge learned in the introductory course and see how those concepts are actually incorporated in the Windows operating system. Student teams work on adding extensions to the operating system and gain valuable software engineering experience.

Topics covered

The Windows Operating System internals provides a fundamental basis for the lectures, but this material is highly augmented with the engineering fundamentals that went into designing and building the system.

Course structure

Class is held in the lab 2 days a week for 1½ hours. The class uses a dedicated lab that the students have 24 hour access to.

ABET Outcomes Assessed

- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to, engage in life-long learning
- (j) knowledge of contemporary issues

Additional ABET Outcomes Covered

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

CSE481B: Tablet PC Capstone (5)

Catalog description

Students work in substantial teams to design, implement, and release a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product. Teams are expected to develop a work plan, and to track and document their progress against it.

Prerequisites

Senior standing in CSE or permission of the instructor.

Specific description

Students work in substantial teams to design, implement, and release a software project involving tablet PCs.

Textbook(s) and/or other required material

Software development environment (Visual Studio), Tablet PC SDK, project support software (e.g., SVN for version control).

Optional text: Building Tablet PC Applications

Course objectives

To learn about the software design process through hands-on development of a software product.

To experience working in larger teams than you have had to deal with previously in our curriculum.

To experience building sophisticated applications by making use of real-world tools, rather than trying to build everything from scratch.

To gain experience dealing with the usability issues related to mobile devices

To have some fun (by building a cool application).

To develop a portfolio documenting your efforts that could be useful in looking for a job.

To gain experience in demonstrating and promoting a prototype application

Topics covered

Software development in teams; basics of pen based computation. Most student time is spent in the development process, and performing critiques of it.

Course structure

Class sessions are divided between regular lecture (especially early in the term), guest lectures, student presentations, and group meetings. Students do four short homework assignments early in the term to get some basic experience in Tablet PC programming, and also do three short writing assignments. The main part of the course is the group project. Each team gives three presentations during the term at check points, as well as a presentation at the final demo. The instructors meet with all student groups following each check point.

ABET Outcomes Assessed

(b) an ability to design and conduct experiments, as well as to analyze and interpret data

(c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(d) an ability to function on multi-disciplinary teams

(e) an ability to identify, formulate, and solve computer engineering problems

(f) an understanding of professional and ethical responsibility

(g) an ability to communicate effectively

(h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context

(i) a recognition of the need for, and an ability to, engage in life-long learning

(j) knowledge of contemporary issues

Additional ABET Outcomes Covered

(a) an ability to apply knowledge of mathematics, science, and engineering

(k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

CSE481C: Robotics Capstone (5)

Catalog description

Students work in substantial teams to design, implement, and release a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product. Teams are expected to develop a work plan, and to track and document their progress against it.

Prerequisites

Senior standing in CSE or permission of the instructor.

Specific description

This capstone course focuses on robotics projects in which groups of 3-4 students program robots to perform a variety of tasks. Four-legged AIBO robots and a blimp were used as test beds in previous offerings.

Textbook(s) and/or other required material

Probabilistic Robotics.
S. Thrun, W. Burgard, and D. Fox.
MIT Press, Cambridge, MA, September 2005.
ISBN 0-262-20162-3.

Course objectives

This course will teach students to understand the key concepts underlying autonomous systems interacting with the real world. By implementing and applying different approaches, the students will learn how to model and control real world systems using probabilistic methods. The programming component of this course will enable the students to solve large scale, open-ended problems in a team setting.

Topics covered

Robot control, probabilistic sensor models, Bayesian state estimation (Kalman and particle filters), robot localization and mapping, path planning, multi-robot coordination.

Course structure

The course consists of lectures and labs, during which the students will acquire the key concepts underlying probabilistic robotics and discuss these methods in the context of the hardware systems used in the course. The students will also present their project work during several lab meetings. The overall system will be demonstrated to the department at the end of the quarter.

ABET Outcomes Assessed

- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to, engage in life-long learning
- (j) knowledge of contemporary issues

Additional ABET Outcomes Covered

- (a) an ability to apply knowledge of mathematics, science, and engineering
 - (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice
-

CSE481D: Games Capstone (5)

Catalog description

Students work in substantial teams to design, implement, and release a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product. Teams are expected to develop a work plan, and to track and document their progress against it.

Prerequisites

Senior standing in CSE or permission of the instructor.

Textbook(s) and/or other required material

Use of development software, typically: Visual Studio, Visual Source Safe, a commercial game engine (e.g., from LithTech), 3D modeling and animation tools (e.g., 3D Studio Max).

Course objectives

To learn about the software design process through hands-on development of a software product.

To experience working in larger teams than you have had to deal with previously in our curriculum.

To experience building sophisticated applications by making use of real-world tools (e.g., the DirectX SDK), rather than trying to build everything from scratch.

To gain experience dealing with the performance demands of high-performance, real-time, distributed applications.

To have some fun (by building a game).

To develop a portfolio documenting your efforts that could be useful in looking for a job.

Topics covered

Software development in teams; basics of games development. Most student time is spent in the development process, and performing critiques of it.

Course structure

3 hours of lectures per week by the instructor and guest lecturers from industry. However, lectures usually are given only during the first two to three weeks. After that, lectures are replaced with meetings with individual teams.

ABET Outcomes Assessed

- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to, engage in life-long learning
- (j) knowledge of contemporary issues

Additional ABET Outcomes Covered

- (a) an ability to apply knowledge of mathematics, science, and engineering
 - (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice
-

CSE481E: UrbanSim Capstone (5)

Catalog description

Students work in substantial teams to design, implement, and release a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product. Teams are expected to develop a work plan, and to track and document their progress against it.

Prerequisites

Senior standing in CSE or permission of the instructor.

Specific description

Students undertake substantial software projects in the context of the UrbanSim land use, transportation, and environmental modeling system.

Textbook(s) and/or other required material

Alan Borning, Paul Waddell, and Ruth Förster, UrbanSim: Using Simulation to Inform Public Deliberation and Decision-Making. To appear, Digital Government: Advanced Research and Case Studies, Hsinchun Chen et al. (eds.), Springer-Verlag, in press.

Edward Beimborn and Rob Kennedy, Inside the Blackbox: Making Transportation Models Work for Livable Communities, Environmental Defense Fund, 1996.

Opus/UrbanSim User Manual.

Course objectives

To learn about the software design process through hands-on development of a software product. This includes the full life cycle of specification, implementation, testing, documentation, and presentation. To experience working in a team, in many cases an interdisciplinary one with CSE students, and students from Civil Engineering, Urban Design and Planning, and other departments.

Even those students who work in a CSE-only team will have substantial interactions with the planning students in lectures and discussions. To gain experience with building software in a complex application domain.

Topics covered

Agile software development processes; simulation; use of models in the planning process; array-based computation. Additional topics will vary by project, and may include human-computer interaction, user testing, Value Sensitive Design, and data mining.

Course structure

Introductory lectures on urban simulation, Python, and scientific computing in Python.

Two assignments, reading background papers, and a major course project. There will be midpoint and final project presentations, and a written report.

ABET Outcomes Assessed

- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to, engage in life-long learning
- (j) knowledge of contemporary issues

Additional ABET Outcomes Covered

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

CSE481F: Computational Biology Capstone (5)

Catalog description

Students work in substantial teams to design, implement, and release a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product. Teams are expected to develop a work plan, and to track and document their progress against it. Computational Biology capstone.

Prerequisites

Senior standing in CSE or permission of the instructor.

Specific description

Students work in teams to design and implement a software tool or software analysis for an important problem in computational molecular biology.

Textbook(s) and/or other required material

none

Course objectives

Students will design, implement, and test a software system to perform high throughput analysis of a problem in molecular biology. Students will work in small teams to produce this software.

Topics covered

Basics of molecular biology, biological sequence analysis, current computational tools and databases for computational molecular biology.

Course structure

Lectures, large team software project, written team project report, team project presentation.

ABET Outcomes Assessed

- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to, engage in life-long learning
- (j) knowledge of contemporary issues

Additional ABET Outcomes Covered

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice

B.4 Other CSE Majors Courses

CSE431: Introduction to Theory of Computation (3)

Catalog description

Models of computation, computable and noncomputable functions, space and time complexity, tractable and intractable functions.

Prerequisites

CSE 322.

Textbook(s) and/or other required material

Michael Sipser, Introduction to the Theory of Computation, 2nd edition, International Thompson Publishing, 2006.

Course objectives

Develop the concepts and skills necessary to be able to evaluate the computability and complexity of practical computational problems.

Topics covered

Turing machines (deterministic, nondeterministic, multitape)
Church-Turing Thesis
Decidability and undecidability, diagonalization, and reducibility
Halting problem, Post correspondence problem, Rice's Theorem, and other undecidability results
Time and space complexity
P vs. NP, NP-completeness, Cook's Theorem, and other NP-complete problems
PSPACE, PSPACE-completeness, PSPACE-complete problems
L vs. NL, NL-completeness, Savitch's Theorem, Immerman-Szelepcsenyi Theorem

Course structure

Three 50-minute lectures per week.
Weekly written assignments.
Midterm and final exam.

CSE450: Animation Production Seminar (1)

Catalog description

Open to all students who have an interest in digital animation. Reviews and analyzes films, animated feature films, and television commercials. Emphasizes the technical and aesthetic basics of animation production in industry studio environments.

Prerequisites

none

Textbook(s) and/or other required material

none

Course objectives

To bring in speakers from the animation and game industries to speak to the students about topics that are of interest to those who wish to get a job in film or game production.

To apply the content covered in the lectures to the specific interest of the student.

Topics covered

Lectures are weekly and generally follow the sequence of the production pipeline.

Speakers show production shot breakdowns as well as research and tool development for the animation industry. Speakers come in who represent both the artistic and technical ends of the spectrum in order to give the students in the class a broad exposure to the reality of working in this challenging and creative environment.

Course structure

Weekly lectures from visiting speakers. One paper required in order to get credit. The paper will cover a creative application of material that was covered in any of the lectures.

CSE455: Computer Vision (4)

Catalog description

Introduction to image analysis and interpreting the 3D world from image data. Topics may include segmentation, motion estimation, image mosaics, 3D-shape reconstruction, object recognition, and image retrieval.

Prerequisites

CSE 303; CSE 326; recommended: MATH 308.

Textbook(s) and/or other required material

No required text. Recommended: Shapiro and Stockman, Computer Vision, Prentice Hall, 2001 or Sonka, Hlavac, and Boyle, Image Processing, Analysis, and Machine Vision, ITP, 1999.

Lecture notes and some technical paper are posted on the class web site.

Course objectives

Students learn the basics of computer vision and some of the state-of-the-art techniques. They will be able to write programs that can perform image segmentation, image matching, object detection or recognition, and applications such as content-based image retrieval or construction of panoramas.

Upon completion of the course they should be able to take an internship or job with a vision company or research lab doing vision or to participate in undergraduate research leading to potential graduate level research.

Topics covered

- * feature detection, descriptors, and matching
- * image segmentation
- * motion
- * mosaics
- * 3D sensing and reconstruction
- * object recognition

Course structure

2 lecture/discussion periods per week. 4 programming projects, each requiring a short report. 2 exams.

CSE456: Story Design for Computer Animation (4)

Catalog description

Animation principles and production for story development and design. Design, development, and production of several storyreels, which are a tool for the pre-production of animated features and shorts. Student use authoring tools to present finished work.

Prerequisites

none

Textbook(s) and/or other required material

The Writer's Journey, Chris Vogler

Course objectives

To teach story structure for short animated film. To develop collaborative skills. To learn cinematography and the technology related to storyreel development. To learn about timing and improv acting and how this relates to short film production for animated films. To understand technical troubleshooting for short animated films. To learn how to plan and create technical production notebooks for short animated films. To learn to use the technology to create storyreels for animated productions. To learn about new techniques for animation production.

Topics covered

Story Structure
Story Structure for Animated Films
Applying story structure techniques to analyze and improve the impact of animated shorts.
Improv and Reference
Physical Humor
Cinematography
Layout and Design
Scoring an animated film
Pitching stories
Responding to feedback from a group.
Planning for Motion
Motion principles
Storyboarding
Character Design
Storyreel Creation

Course structure

Lectures and labs are held weekly and three assignments will be completed by each student. Students will work on teams and select roles that most appeal to them. Written reports will be sent to the teacher with the completed projects.

CSE457: Computer Graphics (4)

Catalog description

Introduction to computer image synthesis, modeling, and animation. Topics may include visual perception, displays and framebuffers, image processing, affine and projective transformations, hierarchical modeling, hidden surface elimination, shading, ray-tracing, anti-aliasing, texture mapping, curves, surfaces, particle systems, dynamics, character animation, and animation principles.

Prerequisites

CSE 303; CSE 326; recommended: MATH 308.

Textbook(s) and/or other required material

Edward Angel, Interactive Computer Graphics: A top-down approach with OpenGL, Fourth Edition.
Addison Wesley, 2005.

Course objectives

Introduction to computer image synthesis and interactive computer graphics applications. Learn fundamentals of 2D and 3D computer graphics modeling, rendering, and animation through homework and projects.

Topics covered

Topics may include visual perception, displays and framebuffers, image processing, affine and projective transformations, hierarchical modeling, hidden surface elimination, shading, ray-tracing, anti-aliasing, texture mapping, curves, surfaces, particle systems, dynamics, character animation, and animation principles.

Course structure

Three hours of lecture
One hour of lab help session (4 times per quarter)
Four projects to be completed, two weeks each, in lab or at home
Two written homeworks

CSE458: Computer Animation (5)

Catalog description

Introduction to basic principles of computer generated animation. Focus on the modeling and lighting of animated characters. Students from Art, CSE, and Music team up on projects to be built on commercially-available modeling and lighting packages.

Prerequisites

either CSE 457, ART 380, or MUSIC 403.

Textbook(s) and/or other required material

Michael O'Rourke, Three Dimensional Computer Animation

Course objectives

To introduce students to the fundamentals of 3D modeling, shading, lighting, animating and rigging characters for three-dimensional computer generated environments.

To understand the complex technical and aesthetic components of the design of animation.

Topics covered

Modeling digital objects that one can find reference for in the real world.

Modeling hard surface and characters for 3D animated digital environments

Shading objects.

Lighting concepts from the real world applied to digital 3D environments

Character Animation Principles

Character Animation Projects

Theory and fundamentals of character rigging for computer animation learning the basics of the animation pipeline for film production

Course structure

Tuesday and Thursday 10:30 - 11:50 Lecture

Sieg 322 and Sieg 325

CSE459: Pre-Production for Collaborative Animation

(5)

Catalog description

Pre-production of collaboratively designed animated shorts. In-depth analysis of classical and computer generated works. Character design and pre-planning, model sheets, character rigging, storyreel and animatics, character motion, design for multiple characters, and principles of animation as applied to character motion and effects.

Prerequisites

CSE 458.

Textbook(s) and/or other required material

none

Course objectives

To be able to design and produce all of the materials needed for the pre-production of an animated short film. To be able to create concept art, reference material, animatics and signature shots that properly support a story. To work on a production team. To meet production deadlines and to troubleshoot technical and aesthetic problems.

Topics covered

Story development for animated shorts. Small team production with an emphasis on the technical and aesthetic pipeline. New tools for character modeling and setup, storyboard and layout, compositing for short films, performance and animation, animation planning, character rigging and character animation exercises. Signature shots and signature shot production.

Course structure

Production teamwork and hands on projects. Four individual production projects and two team based productions will be completed by the students in this class. Students will be expected to present their production work to the rest of the class a minimum of three times during the quarter. This class includes weekly lectures, weekly lab sessions, and written reviews.

CSE460: Animation Capstone (5)

Catalog description

Apply the knowledge gained in previous animation courses to produce a short animated film. Topics include scene planning, digital cinematography, creature and hard surface modeling, animatics and basics of character animation, and rendering techniques.

Prerequisites

CSE 458, CSE 459.

Textbook(s) and/or other required material

none

Course objectives

Students will work together to produce a short animated film using the storyreel, animatic, concept art and signature shots designed and produced in cse459. Students will work on several teams and take on leadership of one of the teams. The production will be required to spend considerable time applying all of the previous skills learned in 456, 458 and 459. Students will need to meet deadlines and work well in a group as every part of the production pipeline will need to succeed in order to create a fully completed film. Students will also create a poster and DVD to present the work that they've completed. Students will take part in screening the film to the campus community.

Topics covered

advanced lighting, advanced efx, advanced animation, advanced shading/texture, advanced cinematography, render quality, meeting deadlines, renderfarm maintenance, compositing and post production. Poster and DVD design.

Course structure

Capstone course meetings are designed around the product that will be produced. Written reports are expected from each student. The students meet frequently with the instructor and other students to troubleshoot technical and aesthetic and story related challenges. Each production team is managed by TA's and the students themselves and often there are workshops offered and visitors from industry to review and provide feedback to the student production team.

CSE464: Advanced Topics in Digital Animation (5)

Catalog description

Students design individual animated works for professional quality demo reels. 2- and 3-D animatics, special effects design, advanced character animation techniques, 3-D paint techniques and integration, short design, sequence planning, non-photorealistic rendering options, interactive animation for pre-planning, and advanced production techniques and strategies.

Prerequisites

CSE 458.

Textbook(s) and/or other required material

none

Course objectives

Each student will produce and direct their own animated short project. This short may be used for their demo reel in order to apply for industry positions. Much of the work will be individual and some will be team based.

Topics covered

professional quality demo reels
Direction for animated shorts
Production roles for animated shorts
advanced 2- and 3-D animatics
advanced special effects design
advanced character animation techniques
3-D paint techniques and integration
Shot and sequence planning,
Options for non-photorealistic rendering
Animation for pre-visualization
Advanced production techniques and strategies.

Course structure

Students will select a personal project to work on and design, plan and execute their own animated short film. Lectures will include a weekly review of student work. Team projects are encouraged as integrated into a final product.

CSE468: Very Large Scale Integration (5)

Catalog description

Introduction to CMOS technology and circuit design; implementation of combinational and sequential logic; VLSI design methodologies; CAD tools for layout, simulation, and validation. Students design a VLSI chip using modern CAD tools.

Prerequisites

CSE 370.

Textbook(s) and/or other required material

Digital Integrated Circuits, A Design Perspective (2nd edition), J. M. Rabaey, A. Chandrakasan, and B. Nikolic, Prentice-Hall, 2003.

Course objectives

Students will:

1. Understand the principles underlying transistor (especially MOSFET) operation.
2. Understand the static and dynamic behavior of physical (i.e. silicon) logic gates, including transistor sizing, time constants, loading and delays, fan-in and fan-out, interconnect, etc.
3. Understand the behavior of physical (i.e. silicon) sequential-logic circuits, including clocking (single and two-phase), clock skew, pipelining, memories and memory access, interconnect, etc.
4. Master the basics of integrated-circuit fabrication, including floorplanning, layout, interconnect, and processing.
5. Master the tools of custom IC design, including schematic entry, simulation (analog, static timing, dynamic timing), layout, DRC, and LVS.
6. Know about advanced topics, including MOS scaling, BJTs and differential (CML) logic, high-speed and low-power circuits, etc.

Topics covered

Transistor basics

- * Fermi levels and band diagrams
- * The pn-junction diode
- * The MOS transistor
- * Sub- and above-threshold MOSFET operation
- * Static and dynamic MOSFET behavior

CAD tools for physical IC design

- * Schematic entry
- * Simulation and analysis
- * Physical layout
- * DRC and LVS to verify your design
- * Parasitic extraction
- * Switch-level timing simulation

Inverters and pass transistors

- * Static and dynamic behavior
- * Noise margin, fan-in and fan-out
- * Time constants, logic levels, drive

Static CMOS logic

- * Basic logic gates
- * Static and dynamic behavior
- * Time constants (RC), loading, fan-in and fan-out, delays
- * Transistor sizing

Dynamic CMOS logic

- * Domino, CVSL logic
- * Dynamic behavior
- * Cascading logic stages

Sequential circuits

- * Static latches and flip-flops
- * Dynamic latches and flip-flops

- * Registers
- * Merged logic
- * Clocking (single and two-phase)
- * Clock skew, delays, race conditions

Chip design

- * Floorplanning
- * Power bussing, power and ground bounce
- * Interconnect and delays
- * Clocking and PLLs
- * Yield
- * Design margin
- * Pads

Memories

- * Static RAM
- * Sense amplifiers
- * Address decoders
- * Dynamic RAM
- * Nonvolatile memories

Arithmetic circuits

- * Adders
- * Multipliers

Advanced topics

- * Copper, SOI/SOS, deep submicron
- * Technology scaling
- * Low-power design
- * Bipolar transistors, differential CML
- * High-speed circuits, controlled impedance lines

Course structure

3 hours of lecture per week

3 hours of laboratory per week

Final project including documentation and report

CSE473: Introduction to Artificial Intelligence (3)

Catalog description

Principal ideas and developments in artificial intelligence: theorem proving, problem-solving methods, representation of knowledge, natural language analysis and synthesis, programming languages for artificial intelligence. Not open for credit to students who have completed 415.

Prerequisites

CSE 326; recommended: CSE 341.

Textbook(s) and/or other required material

S. Russell & P. Norvig, "Artificial Intelligence: A Modern Approach", Prentice Hall, 1995. S. Tanimoto, "Introduction to Python for Artificial Intelligence"

Course objectives

Mastery of the fundamental concepts and techniques of artificial intelligence.

Topics covered

Problem solving and search
Knowledge representation and reasoning
Reasoning under uncertainty
Machine learning
Planning

Course structure

Three 50-minute lectures/week.

CSE490: Special Topics in Computer Science and Engineering (1-5)

Catalog description

Lectures, discussions, and possibly labs on topics of current interest in computer science and engineering not covered by other CSE undergraduate courses.

Prerequisites

Permission of Instructor

Textbook(s) and/or other required material

Various

Course objectives

This course number is used for new courses in development, one-time courses on topics not covered in other CSE undergraduate courses, and for other special topics laboratories and seminars.

Topics covered

Various

Course structure

Various

CSE497: Undergraduate Research Seminar (1)

Catalog description

Students prepare and give a public talk on their faculty-sponsored research projects.

Prerequisites

Students must be participating in research with a CSE faculty member

Textbook(s) and/or other required material

none

Course objectives

To allow students to develop their presentation skills and learn about research from a variety of research areas within the department

Topics covered

Various

Course structure

Weekly 1-hour seminar

CSE498: Senior Project (1-9)

Catalog description

A report (and perhaps demonstration) describing a development, survey, or small research project in computer science or an application to another field. Objectives are: (1) integrating material from several courses, (2) introducing the professional literature, (3) gaining experience in writing a technical document, and (4) showing evidence of independent work. Work normally extends over more than one quarter, for a maximum of 6 credits for 498; 9 credits are required for 498H.

Prerequisites

Permission of CSE faculty member

Textbook(s) and/or other required material

none

Course objectives

Training of undergraduates in the techniques and methods of CSE research and in preparation of a written research report.

Topics covered

Various

Course structure

Individual research with a faculty member over an extended period of time leading to a written research report or senior honors thesis.

CSE 499 Reading and Research (1-24)

Catalog description

Available in special situations for advanced computer science majors to do reading and research in field, subject to approval of undergraduate adviser and CSE faculty member. Free elective, but does not replace core course or computer science elective. Credit/no credit only.

Prerequisites

Permission of CSE faculty member and undergraduate advisor

Textbook(s) and/or other required material

none

Course objectives

To allow students to learn, via individualized instruction, about a topic that is not part of the standard curriculum. This is often preparatory to work on a formal senior project or senior thesis.

Topics covered

Various

Course structure

Individual reading and research with a faculty member.

C. Faculty Curriculum Vitae

Alverson, Gail

Lecturer (2006 Ass't)

Education:

- B.S., Computer Science, Queen's University, Ontario Canada, 1996
- M.S., Computer Science, University of Washington, 1988
- Ph.D., Computer Science, University of Washington, 1990

Other Experience:

- I've spent the last 16 years in industry at Cray Inc., working for a Supercomputer company that pushes the state of the art in terms of performance. The last two years I've brought that software engineering knowledge into the classroom during the Spring offering of CSE 403, software engineering.
- Cray Inc.
- September 2005-Present: Senior Engineering Manager reporting to the VP of Engineering. Responsibilities evolve around technical task management, including running project review forums, technical training forums, and improving project management practices and communication.
- March 2004 - September 2005: Senior Engineering Manager responsible for the message passing toolkit (MPI and SHMEM libraries) across all Cray US products, the Eldorado product runtime library, and third party compiler vendor relationships. Management included technical direction, scheduling, resourcing, and interfacing with groups internal and external to Cray. Point person for all aspects of Cray XT3 programming environments and legal issues.
- Major accomplishments: Negotiated XT3 SHMEM contract with the Department of Energy and delivered ontime; delivered XT3 programming environment on schedule and Rainier MPI and Eldorado Runtime milestones ontime.
- June 2002 - March 2004. Senior Engineering Manager responsible for the compute node operating system and programming environment of Cray's XT3 system for Sandia National Laboratory, named Red Storm. This was a \$90M project, to ship a machine with 10K processors. Led a team focussed on OS and PE components, and managed a number of vendor and customer negotiations.
- Major accomplishments: Completed software architecture to customer approval and satisfied contract milestones; negotiated a PGI reseller agreement for compilers.
- January 2002 - June 2002. MTA Platform Specialist responsible for providing competitive information, influencing product direction, interfacing with sales, training, customers, and engineers.
- Major accomplishments: Established a rich set of marketing collateral and training material, and facilitated a data-mining direction through a professional services agreement and a proposal for a center for data analysis.
- January 2000 - January 2002. Senior Engineering Manager responsible initially for the programming environment for the MTA product and then programming tools and libraries across all Cray products. [This was the period when Seattle-based Tera acquired Minnesota-based Cray Inc. and X1 product, and moved from silo to matrix organization.]
- Major accomplishments: Negotiated the C++ library vendor agreement. Kept the programs and engineers moving forward during the acquisition process and organizational change.
- May 1994 - January 2000. Software Engineering Manager, leading and developing MTA debugger, library, and tools projects.
- Major accomplishments: Team completed and released the debugger, runtime library, and two GUI performance tools.
- July 1991 - May 1994. Software Engineer responsible for the development of the MTA Runtime system library. This work started with low level coding of the user level trap handlers, moved to the development of a thread package, swap management, and an event logging system, and ended with the development of the gdb-based debugger and its interface with the runtime and compiler.
- Major accomplishments: Completed the initial Runtime library. Demonstrated through design papers and implementation technical ability that spanned compilers, libraries, and operating systems.

- University of Washington
- Spring quarter, 2006 and 2007
- Lecturer, teaching the CS 403 undergraduate software engineering class. Responsibilities included preparing and delivering lectures, developing exercises and exams, and evaluating student performance.
- Spring quarter 2001
- Acting Assistant Professor, teaching the CS 341 undergraduate programming language class, which included preparing and delivering lectures, developing exercises and exams, and evaluating student performance.

Consulting, patents, etc.:

- Patents:
- Stream management in a multithreaded environment [7191444]
- Restricting access to memory in a multithreaded environment [7165150]
- Techniques for reducing the rate of instruction issuance [7020767]
- User program and operating system interface in a multithreaded environment [6952827]
- Synchronization techniques in a multithreaded environment [6862635, 7117330]
- Debugging techniques in a multithreaded environment [6848097, 6480818]
- Techniques for an interrupt-free operating system [6314471]

States in which registered: none

Level of activity: professional societies - none; research - medium; consulting/summer work in industry - high.

Anderson, Richard

Professor (1986 Ass't, 1991 Assoc, 1998 Full)

Education:

- B.A., Mathematics, Reed College, 1981
- Ph.D., Computer Science, Stanford, 1986

Other Experience:

- Mathematical Sciences Research Institute, Berkeley, California. Postdoctoral research fellow, 1985-1986
- Visiting Professor, Indian Institute of Science, Bangalore, India, 1993-1994
- Visiting Researcher, Learning Sciences and Technology Group, Microsoft Research, Redmond, Washington. June 2001 - September 2002

Consulting, patents, etc.:

- Design Intelligence, 1997-1999

Principal publications of last five years:

- Anderson, R. J., Anderson, R. E., Hoyer, C., Prince, C., Su, J., Videon, F., and Wolfman, S., "A Study of Diagrammatic Ink in Lecture," *Computers and Graphics*, 29, 2005, pp. 480-489.
- Deibel, K., Anderson, R. J., and Anderson, R. E., "Using Edit Distance to Analyze Cardsorts," *Expert Systems*, 22 (3), 2005, pp. 129-138.
- Anderson, R. J., "The Role of Experiment in the Theory of Algorithms", *Dimacs Series in Discrete Mathematics and Theoretical Computer Science*, Vol 59, 2002, pp. 191-195.
- Anderson, R. J., Kannan, S., Karloff, H., Ladner, R. E., "Thresholds and Optimal Binary Tree Comparison Search Trees," *Journal of Algorithms*, Vol. 44, 2002, 338-358.
- Razmov, V., and Anderson, R., "Experiences with Agile Teaching in Project-based Courses, ASEE Annual Conference, 2006.
- Razmov, V., and Anderson, R., "Pedagogical techniques supported by the use of student devices in teaching software engineering," 37th ACM SIGCSE Technical Symposium on Computer Science Education, 2006, pp. 344-348.
- Ginat, D., Anderson, R., Garcia, D. D., and Rasala, R., "Randomness and Probability in the Early CS Courses," Thirty-sixth SIGCSE Technical Symposium on Computer Science Education, pp. 556-557.
- Anderson, R. J., Anderson, R. E., Hoyer, C., Prince, C., Su, J., Videon, F., and Wolfman, S., "Understanding Diagrammatic Ink in Lecture," *Computers and Graphics*, AAAI Workshop on Ink Understanding.
- Anderson, R. J., Hoyer, C., Prince, C., Su, J., Videon, F., and Wolfman, S., "Speech, Ink, and Slides: The Interaction of Content Channels", *ACM Multimedia* 2004, pp. 796-803.
- Anderson, R. J., Anderson, R., Hoyer, C., and Wolfman, S. A., "A Study of Digital Ink in Lecture Presentation," *CHI 2004: Conference on Human Factors in Computing*, pp. 567-574, 2004.
- Anderson, R. J., Anderson, R., Simon, B., Wolfman, S. A., VanDeGrift, T., and Yasuhara, K., "Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses," *SIGCSE 2004: Technical Symposium on Computer Science Education*, pp. 56-60, March 2004.
- Anderson, R. J., Beavers, J., VanDeGrift, T., and Videon, F., "Videoconferencing and Presentation Support for Synchronous Distance Learning," 33rd ASEE/IEEE Frontiers in Education Conference, Boulder, Colorado, 2003.
- Anderson, R. J., Anderson, R., VanDeGrift, T., Wolfman, S. A., and Yasuhara, K., "Promoting Interaction in Large Classes with Computer Mediated Feedback," *CSCL 2003: Computer Supported Collaborative Learning*, pp.~119-123, Bergen, Norway, 2003.
- Anderson, R. J., Anderson, R., VanDeGrift, T., Wolfman, S. A., and Yasuhara, K., "Interaction Patterns

with a Classroom Feedback System: Making Time for Feedback," CHI 2003, Interactive Poster. Conference on Human Factors in Computing.

- VanDeGrift, T., and Anderson, R. J., "Learning to Support the Instructor: Classroom Assessment Tools as Discussion Frameworks in CS 1," Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education, 2002, pp. 19--23.

Scientific and professional societies of which a member:

- ACM

Honors and awards:

- 2007 UW College of Engineering Teaching Innovator Award
- Best Paper Award, 1996 ACM Symp. on User Interface Software and Technology
- Fulbright Senior Scholar Award, 1993.
- NSF PYI, 1987-1992.
- NSF Graduate Fellowship, Stanford University, 1981-1984.
- Phi Beta Kappa, Reed College, 1981.

Institutional and professional service in last five years:

- 2006-2007 Executive Committee, Teaching Assignments, Course Scheduling, Introductory Computing Stewardship Committee, Undergraduate Diversity Chair, Educational Outreach. Associate Chair for Education
- 2005-2006 Executive Committee, Teaching Assignments, Course Scheduling, Introductory Computing Stewardship Committee, Undergraduate Diversity Chair, Educational Outreach, Associate Chair for Education
- 2004-2005 Executive Committee, Teaching Assignments, Course Scheduling, Introductory Computing Stewardship Committee, Undergraduate Diversity Chair, Educational Outreach. Associate Chair for Education
- 2003-2004 Executive Committee, Engineering council on educational policy. College council on Tri-campus policy, Course assignments, Course Scheduling, Introductory Programming Oversight. Community College Articulation. Educational Outreach Liason.
- 2002-2003 Executive Committee, Educational Technology and the Professional Master's Program, Community College Articulation, Intro programming review.

States in which registered: None

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Anderson, Ruth

Lecturer (2006 Ass't)

Education:

- Ph.D., Computer Science, University of Washington, 2006
- M.S., Computer Science, University of Washington, 1994
- B.S., Computer Science, University of North Carolina, Chapel Hill, 1991

Other Experience:

- Lecturer, University of Washington, Dept. of Computer Science & Engineering, Seattle, WA. (Winter 2006) Taught undergraduate Computer Science course in data structures.
- Teaching Faculty, Lecturer, University of Virginia, Dept. of Computer Science, Charlottesville, VA. (Fall 2000 - Spring 2005) Taught various undergraduate Computer Science courses. Full responsibility for everything associated with courses in data structures, assembly language, introductory programming for majors and non-majors, and programming languages. Prepared and delivered lectures, labs, assignments, exams, and managed course staff of up to 20 teaching assistants. Included teaching in C++, Java, Prolog, Smalltalk, Scheme, and x86 assembly.
- Programmer, IBM Research, T. J. Watson Research Center, Hawthorne, NY. (Summer 1991) Worked in Experimental Software Development Center. Ported and developed telecommunications software for IBM RS/6000.
- Programmer, IBM Corporation, Research Triangle Park, NC. (Summers 1989 & 1990) Developed telecommunications product for VM operating system. Wrote design documents and helped manage two library systems.

Principal publications of last five years:

- Supporting Active Learning and Example Based Instruction with Classroom Technology. Richard Anderson, Ruth Anderson, K. M. Davis, Natalie Linnell, Craig Prince and Valentin Razmov. SIGCSE 2007: Technical Symposium on Computer Science Education, Covington, KY, USA. March 2007.
- Classroom Presenter - A Classroom Interaction System for Active and Collaborative Learning. Richard Anderson, Ruth Anderson, Oliver Chung, K. M. Davis, Peter Davis, Craig Prince, Valentin Razmov and Beth Simon., Workshop on the Impact of Pen-Based Technology on Education (WIPTE 2006), Purdue University, West Lafayette, IN, April 2006.
- Use of Classroom Presenter in Engineering Courses. Richard Anderson, Ruth Anderson, Luke McDowell, Beth Simon. Proceedings of 35th ASEE/IEEE Frontiers in Education Conference (FIE 2005), Indianapolis, IN, U.S.A., October 2005.
- A Study of Diagrammatic Ink in Lecture. Richard Anderson, Ruth Anderson, Crystal Hoyer, Craig Prince, Jonathan Su, Fred Videon and Steve Wolfman. Computers and Graphics (special issue on Pen Computing). Volume 29, Issue 4, pp. 480-489 (August 2005).
- Preliminary Experiences with a Tablet PC Based System to Support Active Learning in Computer Science Courses. Beth Simon, Ruth Anderson, Crystal Hoyer, Jonathan Su, ITiCSE 2004: Proceedings of the 9th Annual Conference on Innovation and Technology in Computer Science Education, pp. 213-217, Leeds, UK. June 2004.
- A Study of Digital Ink in Lecture Presentation. Richard J. Anderson, Ruth Anderson, Crystal Hoyer, Steven A. Wolfman. Long Paper, CHI 2004: Human Factors in Computing Systems, pp. 567-574, Vienna, Austria. April 2004.
- Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses. Richard Anderson, Ruth Anderson, Beth Simon, Steven A. Wolfman, Tammy VanDeGrift, and Ken Yasuhara. SIGCSE 2004: Technical Symposium on Computer Science Education, pp. 56-60, Norfolk, VA, USA. March 2004.
- Promoting Interaction in Large Classes with Computer-Mediated Feedback. Richard J. Anderson, Ruth

Anderson, Tammy VanDeGrift, Steven A. Wolfman, and Ken Yasuhara. Short Paper, CSCL 2003: Computer Support for Collaborative Learning, pp. 119-123. Bergen, Norway. June 2003.

- Using Edit Distance to Analyze Card Sorts. Katherine Deibel, Richard Anderson and Ruth Anderson. Expert Systems: The International Journal of Knowledge Engineering and Neural Networks. Volume 22, No. 3, pp. 129-138 (July 2005).
- A multi-institutional investigation of computer science seniors' knowledge of programming concepts. Laurie Murphy, Renee McCauley, Suzanne Westbrook, et al. Expert Systems: The International Journal of Knowledge Engineering and Neural Networks. Volume 22, No. 3, pp. 147-159 (July 2005).
- A multi-institutional, multi-national study of programming concepts using card sort data. Kate Sanders, Sally Fincher, and Dennis Bouvier, et al. Expert Systems: The International Journal of Knowledge Engineering and Neural Networks. Volume 22, No. 3, pp. 121-128 (July 2005)..

Scientific and professional societies of which a member:

- ACM
- ACM SIGCSE
- ACM SIGCHI

Honors and awards:

- ACM Faculty Award, voted best teacher by students, UVa Department of Computer Science (2004)
- Bob Bandes Memorial Award for excellence in teaching, UW Dept. of Computer Science & Eng. (1992)

Institutional and professional service in last five years:

- Scholarship Application Reviewer, Grace Hopper Celebration of Women in Computing 2006. Reviewed student scholarship applications.
- BOF Proposal Reviewer, Grace Hopper Celebration of Women in Computing 2004, Reviewed proposals for "Birds of a Feather" sessions.
- Grant Proposal Reviewer:
- NSF's Course, Curriculum, and Laboratory Improvement (CCLI) program, March 12-13, 2007, Arlington, VA, USA. Reviewed grant proposals submitted to the CCLI program.
- NSF's Course, Curriculum, and Laboratory Improvement (CCLI) program, Adaptation and Implementation (A&I) track (CCLI - A&E) January 22-25, 2003, Arlington, VA, USA.
- Reviewed grant proposals submitted to the CCLI - A & E program.
- Paper Reviewer:
- Workshop on the Impact of Pen-Based Technology on Education (WIPTE) (2006, 2007)
- ACM Journal on Educational Resources in Computing (JERIC) (2000 - present)
- ACM Special Interest Group on Computer Science Education Symposium (SIGCSE) (2003 - present)
- ACM Innovation and Technology in Computer Science Education (ITiCSE) (2002 - present)
- ASEE/IEEE Frontiers in Education Conference (FIE) (2000 - present)

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Anderson, Thomas

Professor (1997 Ass't, 1997 Assoc, 2001 Full)

Education:

- A.B., Philosophy, Harvard University, 1983
- M.S., Computer Science, University of Washington, 1989
- Ph.D., Computer Science, University of Washington, 1991

Other Experience:

- Founder and Interim CEO/CTO, Asta Networks, 2000 - 2001 (on leave from UW).
- Computer Science Division, University of California, Berkeley. Associate Professor, 1996 - 1997.
- Computer Science Division, University of California, Berkeley. Assistant Professor, 1991 - 1996.
- Digital Equipment Corporation Systems Research Center. Research Intern, Fall, 1990.
- GenRad Incorporated (manufacturer of automated test equipment).
- Development Engineer/Senior Development Engineer, 1983 - 1987.

Consulting, patents, etc.:

- Consultant, Intel Research, 2002 - 2003.
- Consultant, Apex v. Cybex, 1998 - 1999; Palm v. Kessel, 2000; VMWare v. Microsoft, 2005.
- Technical Advisory Board Member, Colusa Software, 1995-1996.

Principal publications of last five years:

- T. Anderson and M. Reiter. "GENI Facility Security." GENI Design Document 06-23, September 2006.
- T. Anderson and A. Vahdat (Ed). "GENI Distributed Services." GENI Design Document 06-24, September 2006.
- T. Anderson, D. Blumenthal, D. Casey, D. Clark, D. Estrin, L. Peterson, D. Raychaudhry, J. Rexford, J. Wroclawski. "GENI: Global Environment for Network Innovations. Conceptual Design, Project Execution Plan." NSF Major Research Equipment Facility Construction Proposal, January 2006.
- T. Anderson, L. Peterson, S. Shenker, J. Turner, editors. "Overcoming Barriers to Disruptive Innovation in Networking." Report of NSF Workshop, January 2005.
- "Achieving Convergence-Free Routing Using Failure-Carrying Packets." Proc. of the ACM SIGCOMM 2007 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication, August 2007.
- M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. "Do Incentives Build Robustness in BitTorrent?" Selected as an Award Paper in Proc. Fourth Symposium on Networked Systems Design and Implementation (NSDI), April 2007.
- R. Mahajan, D. Wetherall, and T. Anderson. "Mutually Controlled Routing with Independent ISPs." Proc. Fourth Symposium on Networked Systems Design and Implementation (NSDI), April 2007.
- T. Isdal, M. Piatek, A. Krishnamurthy, T. Anderson. "Leveraging BitTorrent for End Host Measurements." Proc. 2007 Passive and Active Measurement Conference (PAM '07), April 2007.
- H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. "iPlane: An Information Plane for Distributed Services." Proc. Seventh USENIX Symposium on Operating Systems Design and Implementation (OSDI 06), November 2006.
- E. Katz-Bassett, J. John, A. Krishnamurthy, D. Wetherall, T. Anderson. "Towards IP Geolocation using Delay and Topology Measurements." Proc. Internet Measurement Conference (IMC) 2006, October 2006.
- H. Madhyastha, T. Anderson, A. Krishnamurthy, N. Spring, and A. Venkataramani. "A Structural Approach to Latency Prediction." Proc. Internet Measurement Conference (IMC) 2006, October 2006.
- T. Anderson, A. Collins, A. Krishnamurthy, and J. Zahorjan. "PCP: Efficient Endpoint Congestion

- Control." Proc. Third Symposium on Networked Systems Design and Implementation (NSDI), May 2006.
- R. Mahajan, D. Wetherall, and T. Anderson. "Negotiation-Based Routing Between Neighboring ISPs." Proc. Second Symposium on Networked Systems Design and Implementation (NSDI), May 2005.
- L. Peterson, T. Anderson, S. Shenker and J. Turner. "Overcoming the Internet Impasse through Virtualization." IEEE Computer, April 2005, pp. 62–69.
- R. Grimm, J. Davis, E. Lemar, A. MacBeth, S. Swanson, T. Anderson, B. Bershad, G. Borriello, S. Gribble, and D. Wetherall. "System Support for Pervasive Applications." ACM Transactions on Computer Systems, 22(4):421–486, November 2004.

Scientific and professional societies of which a member:

- IEEE
- ACM
- USENIX

Honors and awards:

- ACM Fellow, 2005.
- ACM/SIGOPS Mark Weiser Award, 2005.
- Diane S. McEntyre Award for Excellence in Teaching, 1995.
- NSF Presidential Faculty Fellowship, 1994.
- Alfred P. Sloan Research Fellowship, 1994.
- NSF Young Investigator Award, 1992.
- 2007 USENIX Symposium on Network Systems Design and Implementation, Best Student Paper, for "Do Incentives Build Robustness in BitTorrent?"
- 2005 IEEE Communications Society William R. Bennett Prize, given for "Measuring ISP Topologies with Rocketfuel" as the best original paper published in IEEE/ACM Transactions on Networking in 2004.
- 2003 USENIX Symposium on Internet Technologies and Systems, Best Student Paper, for "Scriptroute: A Distributed Facility for Internet Measurement."

Institutional and professional service in last five years:

- Chair, Distributed Services Working Group, NSF Global Environment for Network Innovations (GENI) Project, 2006 - pres.
- Member, Planning Committee, NSF Global Environment for Network Innovations (GENI) Project, 2005 - pres.
- Steering Committee Chair and Director, PlanetLab Consortium, 2003 - pres.
- Steering Committee Co-Chair, ACM/USENIX Network Systems Design and Implementation Conference, 2002 - pres.
- ProgramCommittee Co-Chair, ACM SIGCOMM'06 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, 2006.

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Bacon, Dave

Assistant professor (2006 Ass't)

Education:

- B.S., Physics, California Institute of Technology, 1997
- B.S., Literature, California Institute of Technology, 1997
- Ph.D., Physics, University of California, Berkeley, 2001

Other Experience:

- 2005-2006 Principal Research Scientist, Department of Computer Science & Engineering, University of Washington
- 2004-2005 Postdoctoral fellow, Santa Fe Institute, Santa Fe, NM
- 2001-2004 Postdoctoral scholar, Institute for Quantum Information, Caltech

Principal publications of last five years:

- D. Bacon, I.L. Chuang, and A.W. Harrow, Efficient Quantum Circuits for Schur and Clebsch-Gordan Transforms. *Physical Review Letters*, 97, 170502 (2006)
- D. Bacon, A.M. Childs, and W. van Dam, Optimal measurements for the dihedral hidden subgroup problem. *Chicago Journal of Theoretical Computer Science*, 2, (2006)
- D. Bacon, A.M. Childs, and W. van Dam, From Optimal Measurement to Efficient Quantum Algorithms for the Hidden Subgroup Problem over Semidirect Product Groups. *Proceedings of the 46th Annual Symposium on Foundations of Computer Science, IEEE* (Los Alamitos, California) 469 (2005)
- B.F. Toner and D. Bacon, The Communication Cost of Simulating Bell Correlations. *Physical Review Letters* 91, 187904 (2003)
- D. Bacon and B.F. Toner, Bell Inequalities with Auxiliary Communication. *Physical Review Letters* 90, 157904 (2003)

Scientific and professional societies of which a member:

- ACM, IEEE, APS

Honors and awards:

- 1998 Outstanding Graduate Student Instructor Award, U.C. Berkeley
- 1997-1998 Department of Physics Fellowship, U.C. Berkeley
- 1995-1997 Carnation Merit Scholarship (full tuition scholarship), Caltech
- 1997 Frederic W. Hinrichs, Jr. Memorial Award for student leadership, Caltech
- 1996 Green Memorial Scholarship for creative research in science, Caltech

Institutional and professional service in last five years:

- Referee for *Physical Review A*, *Physical Review B*, *Physical Review Letters*, *Europhysics Letters*, *Journal of Physics A*, *Journal of Optics B*, *Physica A*, *Foundations of Physics Letters*, *Quantum Information Processing*, *Quantum Information and Computation*, *New Journal of Physics*, *Information and Computation*
- Advisory board, and acting Secretary/Treasurer for the APS topical group on quantum information, concepts and computation (2005)

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Baer, Jean-Loup

Professor (1969 Ass't, 1974 Assoc, 1979 Full)

Education:

- Diplome d'Ingenieur, Electrical Engineering, Universite de Grenoble, France, 1960
- Doctorat 3e Cycle, Computer Science, Universite de Grenoble, France, 1963
- Ph.D., Engineering, UCLA, 1968

Other Experience:

- University of Minnesota Oct. 1984 - Nov 1984 Control Data Corporation Visiting Chair
- University of Paris VI Sept. 1979 - July 1980 Visiting Professor
- UCLA June 1965-Sept. 1969 Acting Assistant Professor of Computer Science (1968 - 1969)

Consulting, patents, etc.:

- ATT, ETS, Intel etc.

Principal publications of last five years:

- P.Crowley and J.-L.Baer ``A Modeling Framework for Network Processor Systems", Proc. 1st Workshop on Network Processors, Jan 2002
- D.Ortega, E.Ayguad, J.-L.Baer, and M.Valero ``Cost-effective Compiler Directed Memory Prefetching and Bypassing" Proc. Int. Conf. on Parallel Architectures and Compilation Techniques (PACT 02)}, Sep 2002
- P.Crowley and J.-L.Baer ``Worst-Case Execution Time Estimation for Hardware-assisted Multithreaded Processors" Proc. 2nd Workshop on Network Processors, Feb 2003
- J.-L.Baer, D.Low, P.Crowley, and N.Sidhwaney ``Memory Hierarchy Design for a Multiprocessor Look-up Engine", Proc. Int. Conf. on Parallel Architectures and Compilation Techniques (PACT 03)}, Sep 2003

Scientific and professional societies of which a member:

- IEEE Computer Society
- ACM (SIGARCH)

Honors and awards:

- Guggenheim Fellowship, 1979-1980
- IEEE (Life) Fellow 1992,
- ACM Fellow 1997.
- IEEE Golden Core Member

Institutional and professional service in last five years:

- Member Program Committee SIGMETRICS 2001, Performance 02
- ACM SIGARCH Board of Directors (1999-2001)
- Area editor Journal of Parallel and Distributed Computing, up to 2003
- Associate Editor JERIC 2000-2003
- Member Advisory Board Intel MRL 2000-04

- Chair Curriculum Committee Development for Ashesi University (2000-2002)
- CSE Exec. Comm (2001-2002)
- CSE Quads Comm Chair (2001-2002)

States in which registered: none

Level of activity: professional societies - low; research - low; consulting/summer work in industry - none.

Balazinska, Magdalena

Assistant professor (2006 Ass't)

Education:

- Ph.D., Computer Science, Massachusetts Institute of Technology, 2006
- M.S., Electrical Engineering, Ecole Polytechnique de Montreal, 2000
- Bachelor of Engineering, Computer Engineering, Ecole Polytechnique de Montreal, 2000

Other Experience:

- January - May 2004 Massachusetts Institute of Technology. Teaching Assistant. 6.170: Laboratory in Software Engineering.
- May - September 2002 IBM Thomas J. Watson Research Center. Intern in the Synchronization and Messaging Department.
- January - July 2000 Motorola Canada Software Center. Software Engineer.
- May - September 1998 Institute for Information Technology, National Research Council of Canada. Intern in the Software Engineering group.
- May - September 1997 Centre de Recherche Informatique de Montreal. Intern in the HCI group.
- May - September 1996 Institute for Information Technology, National Research Council of Canada. Intern in the Visual Information Technology group.

Principal publications of last five years:

- Roxana Geambasu, Magdalena Balazinska, Steven D. Gribble, and Henry M. Levy. HomeViews: Peer-to-peer middleware for personal data sharing applications. In 26th ACM SIGMOD International Conference on Management of Data (SIGMOD), June 2007.
- Magdalena Balazinska, Amol Deshpande, Michael Franklin, Philippe B. Gibbons, Jim Gray, Suman Nath, Mark Hansen, Michael Liebhold, Alexander Szalay, and Vincent Tao. Data management in the worldwide sensor web. IEEE Pervasive Computing, 6(2), March 2007.
- Roxana Geambasu, Tanya Bragin, Jaeyeon Jung, and Magdalena Balazinska. On-demand view materialization and indexing for network forensic analysis. In Proc. of the Third International Workshop on Networking Meets Databases (NetDB'07), April 2007.
- Evan Welbourne, Magdalena Balazinska, Gaetano Borriello, and Waylon Brunette. Challenges for pervasive RFID-based infrastructures. In IEEE PerCom Workshop on Pervasive RFID/NFC Technology and Applications. (PERTEC'07), March 2007.
- Magdalena Balazinska, Hari Balakrishnan, Samuel Madden, and Michael Stonebraker. Fault-Tolerance in the Borealis Distributed Stream Processing System In 24th ACM SIGMOD International Conference on Management of Data (SIGMOD), June 2005.
- Jeong-Hyon Hwang, Magdalena Balazinska, Alexander Rasin, Ugur Cetintemel, Michael Stonebraker, and Stan Zdonik. High-availability algorithms for distributed stream processing. In Proc. the 21st International Conference on Data Engineering (ICDE), April 2005.
- Daniel J. Abadi, Yanif Ahmad, Magdalena Balazinska, Ugur Cetintemel, Mitch Cherniack, Jeong-Hyon Hwang, Wolfgang Lindner, Anurag S. Maskey, Alexander Rasin, Esther Ryvkina, Nesime Tatbul, Ying Xing, and Stan Zdonik. The design of the Borealis stream processing engine. In Proc. of the Second Biennial Conference on Innovative Data Systems Research (CIDR), January 2005.
- Hari Balakrishnan, Magdalena Balazinska, Don Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Eddie Galvez, Jon Salz, Michael Stonebraker, Nesime Tatbul, Richard Tibbets, and Stan Zdonik. Retrospective on Aurora. VLDB Journal, 13(4), December 2004.
- Magdalena Balazinska, Hari Balakrishnan, and Michael Stonebraker. Contract-based load management in federated distributed systems. In Proc. of the First Symposium on Networked Systems Design and Implementation (NSDI), March 2004.
- Magdalena Balazinska and Paul Castro. Characterizing mobility and network usage in a corporate wireless

local-area network. In Proc. of the First International Conference on Mobile Systems, Applications, and Services (MobiSys), May 2003.

- Stan Zdonik, Michael Stonebraker, Mitch Cherniack, Ugur Cetintemel, Magdalena Balazinska, and Hari Balakrishnan. The Aurora and Medusa projects. IEEE Data Engineering Bulletin (invited paper), 26(1), March 2003.
- Nick Feamster, Magdalena Balazinska, Greg Harfst, Hari Balakrishnan, and David Karger. Infranet: Circumventing Web censorship and surveillance. In Proc. of the 11th USENIX Security Symposium, August 2002. Best student paper award.

Scientific and professional societies of which a member:

- Member of ACM, ACM SIGMOD, and IEEE.

Honors and awards:

- Microsoft Research New Faculty Fellow Finalist (2007).
- Rogel Faculty Support Award (2006).
- Microsoft Fellow (2003-2005).
- USENIX Security Symposium: Best student paper award (2002).
- NSERC scholarship for graduate studies (2000-2002).
- FCAR scholarship for graduate studies (2000-2001).
- NSERC scholarship for graduate studies (1998-99).
- Fessenden-Trott scholarship (1995-1998).
- Canada Scholars (scholarship from Canadian government) (1994-1998).
- Ecole Polytechnique de Montreal scholarship (1994-1997).

Institutional and professional service in last five years:

- DMSN 2007 program co-chair.
- Co-chair of the ACM SIGMOD New Researcher Symposium (at SIGMOD 2007).
- Program committee member for VLDB 2007, SIGMOD 2007, ICDE 2007, MobiDE 2006, and NetDB 2006.
- Publicity chair MobiDE 2006 and MobiDE 2007.
- Co-chair of the ACM SIGMOD Life After Graduation Symposium (at SIGMOD 2006).
- Co-founder of NWDS, the Northwest Database Society (January 2006)
- Reviewer for TMC (2007), TPDS (2007), VLDB Journal (2006), IEEE Pervasive computing (2006), SIGMOD'06, NSDI'05, IPTPS'05, SIGMOD'04, HotNets'04, MobiSys'04, Infocom'03, and MC2R (2003).
- Program committee member for the SIGMOD'07 undergraduate scholarship program.
- Engineering Open House 2007. Demonstrating the RFID Ecosystem (April 2007).
- Panelist at the 16th Annual Women in Science and Engineering Conference (January 2007).

States in which registered: none

Level of activity: professional societies - none; research - high; consulting/summer work in industry - none.

Beame, Paul

Professor (1987 Ass't, 1993 Assoc, 1999 Full)

Education:

- B.Sc., Mathematics, University of Toronto, 1981
- M.Sc., Computer Science, University of Toronto, 1982
- Ph.D., Computer Science, University of Toronto, 1987

Other Experience:

- Visiting Associate Professor, University of Toronto, 1993-4.
- Postdoctoral Research Associate, Laboratory for Computer Science, M.I.T., 1986-7.
- Lecturer for course, Discrete Mathematics for Computer Science, University of Toronto, Summer 1985

Consulting, patents, etc.:

- Microsoft Research, 2001-2002.

Principal publications of last five years:

- Paul Beame. Proof complexity. In Steven Rudich and Avi Wigderson, editors, Computational Complexity Theory, volume 10 of IAS/Park City mathematics series, pages 199-246. American Mathematical Society, 2004.
- Paul Beame, Russell Impagliazzo, and Ashish Sabharwal. The resolution complexity of independent sets and vertex covers in random graphs. Computational Complexity, 2007.
- Paul Beame, Toniann Pitassi, Nathan Segerlind, and Avi Wigderson. A strong direct product theorem for corruption and the multiparty communication complexity of set disjointness. Computational Complexity, 2007. Invited submission for special issue on CCC 2005.
- Paul Beame, Henry Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. Journal of Artificial Intelligence Research, 22:319-351, 2004.
- Dimitris Achlioptas, Paul Beame, and Michael Molloy. A sharp threshold in proof complexity yields lower bounds for satisfiability search. Journal of Computer and System Sciences, 68(2):238-268, March 2004. Special issue on STOC 2001 conference.
- Paul Beame, Michael Saks, Xiaodong Sun, and Erik Vee. Time-space trade-off lower bounds for randomized computation of decision problems. Journal of the ACM, 50(2):154-195, 2003.
- Paul Beame and Faith Fich. Optimal bounds for the predecessor problem and related problems. Journal of Computer and System Sciences, 65(1):38-72, August 2002. Special issue of selected papers from 1999 STOC conference.
- Paul Beame, Richard Karp, Toniann Pitassi, and Michael Saks. The efficiency of resolution and Davis-Putnam procedures. SIAM Journal on Computing, 31(4):1048-1075, 2002.
- Conference Publications
- Paul Beame, T. S. Jayram, and Atri Rudra. Lower bounds for randomized read/write stream algorithms. In Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, San Diego, CA, June 2007. To appear.
- Paul Beame, Matei David, Toniann Pitassi, and Philipp Woelfel. Separating deterministic from nondeterministic NOF multiparty communication complexity. In Automata, Languages, and Programming: 34th International Colloquium, ICALP 2007. To appear.
- Tian Sang, Paul Beame, and Henry Kautz. A dynamic approach for MPE and Weighted MAX-SAT. In Proceedings of the 20th International Joint Conference in Artificial Intelligence (IJCAI), pages 173-179, Hyderabad, India, January 2007.
- Tian Sang, Paul Beame, and Henry Kautz. Performing Bayesian inference by weighted model counting. In

Proceedings, AAAI-05: Twentieth National Conference on Artificial Intelligence, pages 475-482, Pittsburgh, PA, August 2005. American Association for Artificial Intelligence.

- Tian Sang, Fahiem Bacchus, Paul Beame, Henry Kautz, and Toniann Pitassi. Combining component caching and clause learning for effective model counting. In SAT 2004, Satisfiability Conference, pages 20-28, 2004.
- Dimitris Achlioptas, Paul Beame, and Michael Molloy. Exponential bounds for DPLL below the satisfiability threshold. In Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 139-140, New Orleans, LA, January 2004.
- Paul Beame, Henry Kautz, and Ashish Sabharwal. On the power of clause learning. In Proceedings of the 18th International Joint Conference in Artificial Intelligence (IJCAI), pages 94-99, Acapulco, Mexico, August 2003.
- Paul Beame, Russell Impagliazzo, Toniann Pitassi, and Nathan Segerlind. Memoization and DPLL: Formula caching proof systems. In Proceedings Eighteenth Annual IEEE Conference on Computational Complexity, pages 225-236, Aarhus, Denmark, July 2003.
- Ashish Sabharwal, Paul Beame, and Henry Kautz. Using problem structure for efficient clause learning. In Proceedings of the Sixth International Conference on Theory and Applications of Satisfiability Testing (SAT 2003), pages 159-166, 2003.

Scientific and professional societies of which a member:

- Association for Computing Machinery
- IEEE Computer Society

Honors and awards:

- Presidential Young Investigator Award, National Science Foundation, 1988
- University of Toronto Open Fellowship, 1985-6
- Natural Sciences and Engineering Research Council Postgraduate Scholarship, 1981-5

Institutional and professional service in last five years:

- Vice-Chair, Technical Committee on Mathematical Foundations of Computing, IEEE Computer Society, 2002-present.
- Conference Co-chair, ACM Symposium on Theory of Computing, STOC 2006.
- Program Committee Member, International Computer Science Symposium in Russia, CSR 2006.
- Program Committee Member, IEEE Symposium on Foundations of Computer Science, FOCS 2005.
- Program Committee Member, IEEE Symposium on Foundations of Computer Science, FOCS 2003.
- Program Committee Member, International Workshop on Randomization and Computation, RANDOM 2003.
- NSF CCF-TOC Panelist, 2006.
- Program Advisory Committee, Newton Institute special half-year on Logic and Algorithms, Cambridge University, 2006.
- Associate Editor, Computational Complexity 1997-2003.
- Review Committee, Department of Mathematics, University of Washington, 2005.
- University of Washington Faculty Senate, 2002-6.
- ACMS Steering Committee, 2003-present.

States in which registered: none

Level of activity: professional societies - high; research - high; consulting/summer work in industry - low.

Bershad, Brian

Professor (1993 Ass't, 1996 Assoc, 2006 Full)

Education:

- Ph.D., Computer Science, University of Washington, 1990
- M.S., Computer Science, University of Washington, 1990
- B.S., Electrical Engineering and Computer Science, University of California, Berkeley, 1986

Other Experience:

- Appliant, Inc. Founder (1997), and CEO (2000-2001)
- University of Washington, Department of Computer Science & Engineering, Research Assistant, 1986-1990
- Digital Equipment Corporation Systems Research Center, Research Intern, January 1988 to April 1988; June 1988 to September 1988
- University of California Berkeley, Department of Electrical Engineering and Computer Science, Assistant Programmer, 1984-1986
- Brian R. Demsey & Associates, Principal Programmer for actuarial firm/software vendor, 1981-1984

Consulting, patents, etc.:

- Member, Technical Advisory Board: Rendition Networks, 2001-present; Lockdown Networks, 2002-present
- Member, Board of Directors: Appliant, Inc. (Chairman), 1997-2002; RemoteNet, Inc., 1997-present
- Patents:
- Process for rewriting executable content on a network server or desktop machine in order to enforce site-specific properties, US Patent 6865735.
- Process for transparently enforcing protection domains and access control as well as auditing operations in software components, US Patent 6317868, granted 11/13/2001. (The University of Washington licensed this patent to Microsoft in 2004)..

Principal publications of last five years:

- Mike Swift, Annamalai Muthu, Brian Bershad, Hank Levy. "Recovering Device Drivers." Proceedings of the 2004 Symposium on Operating Systems Design and Implementation, December 2004. Received best paper award. A version of this paper will appear in an upcoming special issue of ACM Transactions on Computer Systems.
- Robert Grimm, David Wetherall, Janet Davis, Eric Lemar, Adam Macbeth, Steven Swanson, Thomas Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble. "System Support for Pervasive Applications." ACM Transactions on Computer Systems, Volume 22, Issue 4, November 2004.
- Paul Gauthier, Brian Bershad, Steven D. Gribble. "Dealing with Cheaters in Anonymous Peer-to-Peer Networks." University of Washington Technical Report 04-01-03, January 2004.
- Mike Swift, Brian Bershad, Henry M. Levy. "Improving the Reliability of Commodity Operating Systems." Proceedings of the 19th Symposium on Operating Systems Principles, November 2003. Received best paper award. A version of this paper will appear in an upcoming special issue of ACM's Transactions on Computer Systems.
- Joshua Redstone, Michael M. Swift, Brian N. Bershad. "Using Computers to Diagnose Computer Problems." Proceedings of the 9th Workshop on Hot Topics in Operating Systems, May 2003.
- Robert Grimm and Brian N. Bershad. "Future Directions: System Support for Pervasive Applications." Proceedings of the International Workshop on Future Directions in Distributed Computing, pages 56–59, Bertinoro, Italy, June 2002. Also appeared in Future Directions in Distributed Computing, pages 212–217,

Scientific and professional societies of which a member:

- Mobisys
- Symposium on Operating System Principles (SOSP)
- Networked Systems Design and Implementation (NSDI)
- Sigmetrics Measurement and Modelling (Sigmetrics)
- Operating Systems Design and Implementation (OSDI)
- Architectural Support for Programming Languages and Operating Systems (ASPLOS)
- International Symposium on Computer Architecture (ISCA)
- Workshop on Workstation Operating Systems (WWOS)
- International Conference on Distributed Computing Systems (ICDCS)
- Advisory Board for National Science Foundation's Office of Polar Programs

Honors and awards:

- 2004: ACM Mark Weiser Award, received in recognition for twenty years of innovation in operating systems research. Past recipients of the award include Frans Kaashoek (MIT), Mendel Rosenblum (Stanford), and Mike Burrows (Google).
- 2004: Best Paper. Mike Swift, Annamalai Muthu, Brian Bershad, Hank Levy. "Recovering Device Drivers." Proceedings of the 2004 Symposium on Operating Systems Design and Implementation, December 2004.
- 2003: Best Paper. Mike Swift, Brian Bershad, Henry M. Levy. "Improving the Reliability of Commodity Operating Systems." Proceedings of the 19th Symposium on Operating Systems Principles, November 2003.
- 2000: Best Paper. Yasushi Saito, Brian Bershad, Hank Levy. "Manageability, Availability and Performance in Porcupine: a Highly Scalable, Cluster-Based Mail Service." 17th Symposium on Operating Systems Principles (SOSP).
- 1994: NSF Presidential Faculty Fellowship (PFF)

Institutional and professional service in last five years:

- Mobisys; 2004.
- Symposium on Operating System Principles (SOSP); 1991, 2003, 2005.
- Networked Systems Design and Implementation (NSDI); 2003.
- Sigmetrics Measurement and Modelling (Sigmetrics); 2003.
- Operating Systems Design and Implementation (OSDI); 1994, 2001. Cochair for OSDI 2006.
- Architectural Support for Programming Languages and Operating Systems (ASPLOS); 1998.
- International Symposium on Computer Architecture (ISCA); 1997.
- Workshop on Workstation Operating Systems (WWOS) IV (Chair); 1993.
- International Conference on Distributed Computing Systems (ICDCS); 1993.
- Member, Advisory Board for National Science Foundation's Office of Polar Programs; 2004-2006.
- Various NSF Peer Review Panels, including the 2005 ITR COV (Committee of Visitors).

States in which registered: none

Level of activity: professional societies - medium; research - medium; consulting/summer work in industry - high.

Borning, Alan

Professor (1980 Ass't, 1987 Assoc, 1993 Full)

Education:

- B.A., Mathematics, Reed College, 1971
- M.Sc., Computer Science, Stanford University, 1973
- Ph.D., Computer Science, Stanford University, 1979

Other Experience:

- Visiting Professor at Ashesi University, Ghana, February-March 2004
- Visiting Professor, University of Hamburg, Germany, October-December 2003
- Visiting Professor at Monash University, Australia, January-February 2001
- Visiting Professor at University of Melbourne and Monash University, Australia (on sabbatical leave from UW) January--July 1997
- Visiting Scientist at Rank Xerox EuroPARC, Cambridge, England (on sabbatical leave from UW), Sept 1989 - Sept 1990
- Postdoctoral Fellow, Department of Artificial Intelligence, University of Edinburgh, 1979 - 1980

Consulting, patents, etc.:

- Consulting:
- Aldus Corporation, Seattle, Washington
- American Bell, Indianapolis, Indiana
- Apple Computer, Cupertino, California
- Atari Sunnyvale Research Center, Sunnyvale, California
- Data I/O, Redmond, Washington
- Intel Corporation, Aloha, Oregon
- Tektronix Computer Research Laboratory, Beaverton, Oregon
- Xerox Palo Alto Research Center, Palo Alto, California

Principal publications of last five years:

- Alan Borning, Paul Waddell, and Ruth Foerster, "UrbanSim: Using Simulation to Inform Public Deliberation and Decision-Making," in Hsinchun Chen et al., *Digital Government: Advanced Research and Case Studies*, Springer-Verlag, in press.
- Janet Davis, Peyina Lin, Alan Borning, Batya Friedman, Peter Kahn, and Paul Waddell, "Simulations for Urban Planning: Designing for Human Values," *IEEE Computer*, Special Issue on Urban Computing, Vol. 39 No. 9, September 2006, pages 66-72.
- Hana Sevcikova, Alan Borning, David Socha, and Wolf-Gideon Bleek, "Automated Testing of Stochastic Systems: A Statistically Grounded Approach," *Proceedings of the International Symposium on Software Testing and Analysis*, ACM, July 2006.
- Batya Friedman, Peter H. Kahn Jr., and Alan Borning, "Value Sensitive Design and Information Systems," in P. Zhang and D. Galletta, *Human-Computer Interaction and Management Information Systems: Foundations*, M.E. Sharpe, Armonk, NY, 2006.
- Alan Borning, Batya Friedman, Janet Davis, and Peyina Lin, "Informing Public Deliberation: Value Sensitive Design of Indicators for a Large-Scale Urban Simulation," *Proceedings of the 9th European Conference on Computer Supported Cooperative Work*, Paris, September 2005.
- Michael Noth, Alan Borning, and Paul Waddell, "An Extensible, Modular Architecture for Simulating Urban Development, Transportation, and Environmental Impacts," *Computers, Environment and Urban Systems*, Vol. 27 No. 2, March 2003, pages 181-203.

- Warwick Harvey, Peter Stuckey, and Alan Borning, "Fourier Elimination for Compiling Constraint Hierarchies," *Constraints: An International Journal*, Vol. 7 No. 2, April 2002, pages 199-219.

Scientific and professional societies of which a member:

- Association for Computing Machinery
- Computer Professionals for Social Responsibility

Honors and awards:

- Faculty Recognition Award, Minority Science and Engineering Program, University of Washington, February 1996 (for work in setting up and maintaining a tutoring program for women and minority students in computer science and engineering).
- Best Paper Award, 1996 ACM Symposium on User Interface Software and Technology. (The paper is Alan Borning, Richard Anderson, and Bjorn Freeman-Benson, "Indigo: A Local Propagation Algorithm for Inequality Constraints.")
- Fulbright Senior Scholar Award for research and teaching in Australia, 1997.
- ACM Fellow, 2001.

Institutional and professional service in last five years:

- Doctoral Program Committee, ACM Conference on Object-Oriented Programming Systems, Languages, and Applications. Conference held November 2002, Seattle.
- Program Committee, Workshop on User-Interaction in Constraint Satisfaction, 9th International Conference on Principles and Practice of Constraint Programming (CP 2003), Cork, Ireland, September 2003.
- Doctoral Consortium, ACM Conference on Computer Human Interaction, April 2006, Montreal, Canada.
- Program co-chair, 8th Annual International Digital Government Research Conference. Conference to be held May 2007, Philadelphia, PA.
- Dept of Computer Science & Engineering, Colloquia and Distinguished Lecturers Chair, 1997--2000.
- Dept of Computer Science & Engineering, Departmental Retreat organizer, 1999.
- Dept of Computer Science & Engineering, Graduate admissions committee, 1997-1998.
- Dept of Computer Science & Engineering, Faculty teaching evaluation coordinator, 1997-1998.
- Dept of Computer Science & Engineering, Curriculum Committee Chair, 2000-2001.
- College of Engineering Research Policy Committee, 1997-1999.
- College of Engineering Educational Policy Committee, Chair, 1995--96; member, 1999-2002.
- CSE Undergraduate Program Committee, 2002-2003, 2005
- CSE Commercialization Oversight Committee, 2005-2007
- CSE Information School Liason, 2002
- UW College of Engineering Council on Educational Policy, 2002-2003, 2004-2006
- UW College of Engineering College Council, 2006-present

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Borriello, Gaetano

Professor (1988 Ass't, 1993 Assoc, 1998 Full)

Education:

- B.S., Electrical Engineering, Polytechnic Institute of New York, 1979
- M.S., Electrical Engineering, Stanford University, 1981
- Ph.D., Computer Science, University of California at Berkeley, 1988

Other Experience:

- Xerox Corporation Palo Alto Research Center, Member of Research Staff, 1981-1984
- Intel Corporation, Founding Director Intel Research Seattle, 2001-2003
- Intel Corporation, Member of Research Staff, 2003-2006

Consulting, patents, etc.:

- Cadence Design Systems, 1998
- Georgia Board of Regents, 1999-2000
- Consystant Design Technologies (Chair, Technical Advisory Board), 1999-

Principal publications of last five years:

- L. Arnstein, G. Borriello, S. Consolvo, C.Y. Hung, J. Su. Labscape: A Smart Environment for the Cell Biology Laboratory, IEEE Pervasive Computing, Vol. 1, No. 3, pp. 13-21, July-September 2002.
- D. Fox, J. Hightower, L. Liao, D. Schultz, G. Borriello. Bayesian Filtering for Location Estimation. IEEE Pervasive Computing (special issue on Dealing with Uncertainty), Vol. 2, No. 3, pp. 24-33, July-September 2003.
- B. Hemingway, W. Brunette, T. Anderl, G. Borriello. The Flock: Using Wireless Mote Networks in an Undergraduate Curriculum. IEEE Computer (special issue on Sensor Networks), Vol. 37, No. 8, pp. 72-78, August 2004.
- R. Grimm, J. Davis, E. Lemar, A. MacBeth, S. Swanson, T. Anderson, B. Bershad, G. Borriello, S. Gribble, D. Wetherall. System Support for Pervasive Applications. ACM Transactions on Computer Systems, Vol. 22, No. 4, pp. 421-486, November 2004.
- G. Borriello, M. Chalmers, A LaMarca, P. Nixon. Delivering Real-World Ubiquitous Location Systems. Communications of the ACM, Vol. 48, No. 3, pp. 36-41, March 2005.
- J. Kang, E. Welbourne, B. Stewart, G. Borriello. Extracting Places from Traces of Locations. ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 9, No. 3, pp. 58-68, July 2005.
- S. Jain, G. Borriello. Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks. Mobile Networks and Applications (MONET) Journal Special Issue on WiOpt 2004, Vol. 11, No. 3, pp. 327-339, June 2006.
- H. Hile, J. Kim, G. Borriello. Microbiology Tray and Pipette Tracking as a Proactive Tangible User Interface. 2nd International Conference on Pervasive Computing, Linz, Austria, pp. 323-339, April 2004.
- J. Lester, B. Hannaford, G. Borriello. "Are You With Me?" – Using Accelerometers to Determine if Two Devices are Carried by the Same Person. 2nd International Conference on Pervasive Computing, Linz, Austria, pp. 33-50, April 2004.
- A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, B. Schilit. Place Lab: Device Positioning Using Radio Beacons in the Wild. 3rd International Conference on Pervasive Computing, Munich, Germany, Springer LNCS 3468, pp. 116-133, May 2005.
- G. Borriello, A. Liu, T. Offer, C. Palistrant, R. Sharp. WALRUS: Wireless Acoustic Location with Room-Level Resolution using Ultrasound. 3rd International Conference on Mobile Systems, Applications, and

Services, Seattle, WA, pp. 191-203, June 2005.

- J. Lester, T. Choudhury, G. Borriello. A Practical Approach to Recognizing Physical Activities. 4th International Conference on Pervasive Computing, Dublin, Ireland, Springer LNCS 3968, pp. 1-16, May 2006.
- J.H. Kang, G. Borriello. Harvesting of Location-Specific Information through WiFi Networks. 2nd International Symposium on Location- and Context-Awareness (LoCA 2006), Dublin, Ireland, Springer LNCS 3987, pp. 86-102, May 2006.
- A. Liu, H. Hile, H. Kautz, G. Borriello, P. Brown, M. Harniss, K. Johnson. Indoor Wayfinding: Developing a Functional Interface for Individuals with Cognitive Impairments. 8th International ACM SIGACCESS Conference on Computers and Accessibility, Portland, OR, pp. 95-102, October 2006.

Scientific and professional societies of which a member:

- IEEE/Computer Society
- ACM/SIGDA
- Computer Professionals for Social Responsibility

Honors and awards:

- Jerre D. Noe Endowed Professorship – Dep't of CSE, University of Washington 2006-
- Divisional Recognition Award – Corporate Technology Group, Intel Corporation 2003
- Fulbright Pisa Chair Award – Scuola Superiore Sant'Anna, Pisa, Italia 1995-1996
- Distinguished Teaching Award – University of Washington 1995
- Junior Faculty Achievement Award – College of Engineering, University of Washington 1994
- Presidential Young Investigator Award – National Science Foundation 1988

Institutional and professional service in last five years:

- NRC/CSTB Study on Radio-Frequency Identification (chair) 2004
- DARPA Information Science and Technology (ISAT) Study Group (member) 2001-2003
- NSF/CISE Advisory Board (member) 2001-2003
- Associate Editor-in-Chief, IEEE Pervasive Computing Magazine 2006-
- Editorial Board Member, IEEE Pervasive Computing Magazine 2001-2005
- International Conference on Distributed Computing Systems – ICDCS (27th: 2007)
- IEEE Workshop on Mobile Computing Systems and Applications – HotMobile (5th: 2006, 6th: 2007, 7th: 2008)
- International Symposium on Location and Context-Aware Computing (2nd: 2006, 3rd: 2007)
- International Conference on Ubiquitous Computing (6th: 2004, 7th 2005, 9th: 2007, late results track)
- ACM Conference on Embedded Networked Sensor Systems (SenSys) (2nd: 2004, 5th: 2007)
- International Conference on Mobile Systems, Applications, and Services (1st: 2003, 2nd: 2004)
- International Conference on Pervasive Computing (1st: 2002, 3rd: 2005)
- International Conference on Architecture of Computing Systems – ARCS (2002, 2004)
- UW/CSE Associate Chair for Research 2006-
- UW/CSE Executive Committee 2004-2006
- UW/CSE Faculty Recruiting Committee 2003-2004, 2004-2007 (chair)
- UW Graduate School Review of Department of Technical Communication (chair) 2006
- UW Exploratory Center for Obesity Research Steering Committee 2005-

States in which registered: None.

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Chambers, Craig

Professor (1991 Ass't, 1997 Assoc, 2003 Full)

Education:

- S.B., Computer Science, MIT, 1986
- Ph.D., Computer Science, Stanford University, 1992

Other Experience:

- 1/07 to 5/07: Visiting Faculty, School of Mathematics, Statistics, and Computer Science, Victoria University of Wellington, New Zealand
- 8/06 to 12/06: Visiting Scientist, Google
- 8/99 to 3/00: Visiting Faculty, School of Computer Science, Carnegie Mellon University
- 4/99 to 7/99: Visiting Scholar, IBM T.J. Watson Research Center

Consulting, patents, etc.:

- Craig Chambers, Susan J. Eggers, Brian K. Grant, Markus Mock, and Matthai Philipose. System and Method for Performing Selective Dynamic Compilation Using Run-Time Information. United States Patent #6,427,234. Issued July 30, 2002.

Principal publications of last five years:

- Curtis Clifton, Todd Millstein, Gary Leavens, and Craig Chambers. MultiJava: Design Rationale, Compiler Implementation, and Applications. ACM Transactions on Programming Languages and Systems (TOPLAS), Vol. 23, No. 3, pp. 517-575, May 2006.
- Markus Mock, Darren C. Atkinson, Craig Chambers, and Susan J. Eggers. Program Slicing with Dynamic Points-To Sets. IEEE Transactions on Software Engineering (TSE), Vol. 31, No. 8, pp. 657-678, August 2005. An earlier version, Improving Program Slicing with Dynamic Points-To Data, appeared in Proceedings of the ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE '02), Charleston, SC, November 2002.
- Todd Millstein, Colin Bleckner, and Craig Chambers. Modular Typechecking for Hierarchically Extensible Datatypes and Functions. ACM Transactions on Programming Languages and Systems (TOPLAS), Vol. 26, No. 5, September 2004. Earlier versions appeared in Proceedings of the 2002 International Conference on Functional Programming (ICFP '02), Pittsburgh, PA, October 2002, and Proceedings of the Ninth International Workshop on Foundations of Object-Oriented Languages (FOOL '02), Portland, OR, January 2002.
- Brian Grant, Matthai Philipose, Markus Mock, Craig Chambers, and Susan Eggers. A Retrospective on "An Evaluation of Staged Run-Time Optimizations in DyC." Invited retrospective, 20 Years of the ACM SIGPLAN Conference on Programming Language Design and Implementation (1979 - 1999): A Selection, ACM SIGPLAN Notices, Vol. 39, No. 4, April 2004.
- Craig Chambers and David Ungar. A Retrospective on "Customization: Optimizing Compiler Technology for Self, a Dynamically-Typed Object-Oriented Programming Language." Invited retrospective, 20 Years of the ACM SIGPLAN Conference on Programming Language Design and Implementation (1979 - 1999): A Selection, ACM SIGPLAN Notices, Vol. 39, No. 4, April 2004.
- Optimizer Flow Functions from Semantic Meanings. In Proceedings of the 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '07), San Diego, CA, June 2007. An earlier version was presented at the 2005 International Workshop on Compiler Optimization Meets Compiler Verification (COCV '05), Edinburgh, Scotland, April 2005.
- Keunwoo Lee and Craig Chambers. Parameterized Modules for Classes and Extensible Functions. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP '06), Nantes, France,

July 2006. An earlier version appeared in Proceedings of the 2006 International Workshop on Foundations and Developments of Object-Oriented Languages (FOOL/WOOD '06), Charleston, SC, January 2006.

- ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '03), Anaheim, CA, October 2003.
- Jonathan Aldrich, Vibha Sazawal, Craig Chambers, and David Notkin. Language Support for Connector Abstractions. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP '03), Darmstadt, Germany, July 2003.
- Sorin Lerner, Todd Millstein, and Craig Chambers. Automatically Proving the Correctness of Compiler Optimizations. In Proceedings of the 2003 SIGPLAN Conference on Programming Language Design and Implementation (PLDI '03), San Diego, CA, June 2003. Awarded "Best Paper."
- Kathleen Fisher and Craig Chambers. Experiences with Author Response at PLDI and IFCP 2004. ACM SIGPLAN Notices, Vol. 39, No. 12, December 2004.

Scientific and professional societies of which a member:

- Association of Computing Machinery (ACM)
- ACM Special Interest Group on Programming Languages (SIGPLAN)

Honors and awards:

- 2006: Most Influential Paper Award from PLDI'95
- 2003: Distinguished Lecturer, Department of Computer and Information Science, University of Pennsylvania
- 2003: Best Paper Award, PLDI'03
- 2003: Two papers included in 20 Years of the ACM SIGPLAN Conference on Programming Language Design and Implementation (1979 - 1999): A Selection
- 1994: NSF Young Investigator Award
- 1992: NSF Research Initiation Award

Institutional and professional service in last five years:

- Selection committee, Most Influential Papers at OOPSLA, 1986-1996
- External reviewer, 2007 ACM SIGPLAN History of Programming Languages Conference (HOPL-III)
- Program committee, 2007 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '07)
- Program committee, 2006 Joint Modular Languages Conference (JMLC '06)
- Program committee, 2006 European Conference on Object-Oriented Programming (ECOOP '06)
- Mentor, Doctoral Symposium, 2005 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '05)
- Chair, program committee, 2004 SIGPLAN Conference on Programming Language Design and Implementation (PLDI '04)

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Curless, Brian

Associate professor (1998 Ass't, 2003 Assoc)

Education:

- Ph.D., Electrical Engineering, Stanford University, 1997
- M.S., Electrical Engineering, Stanford University, 1991
- B.S., Electrical Engineering, University of Texas at Austin, 1988

Other Experience:

- Associate Professor, University of Washington, Department of Computer Science & Engineering (2003 - present).
- Visiting Researcher and Consultant, Microsoft Research, Interactive Visual Media Group (2004 - 2005).
- Assistant Professor, University of Washington, Department of Computer Science & Engineering (1998 - 2003).
- Visiting Scientist, Digital Michelangelo Project, Florence, Italy (Winter 1999).
- Scientific Advisory Board Member, Paraform, Inc., Mountain View, CA (1998 - 2002).
- Research Associate, Digital Michelangelo Project, Stanford University, with Professor Marc Levoy (1997).
- Research Assistant, Stanford University, with Professor Marc Levoy (1991 - 1997).
- Software Contractor, Silicon Graphics Inc., Mountain View, CA (1993).
- Research Engineer, Remote Measurements Laboratory at SRI International, Menlo Park, CA (1988 - 1989).

Consulting, patents, etc.:

- Consultant for Adobe, summer 2006.
- Consultant for Microsoft Research, summer 2005.

Principal publications of last five years:

- Schematic storyboards for video editing and visualization. Dan Goldman, Brian Curless, David Salesin, Steve Seitz, SIGGRAPH 2006, Boston, MA, August 2006.
- Multiview stereo revisited. Michael Goesele, Brian Curless, Steve Seitz, CVPR 2006, New York, NY, June 2006.
- Piecewise image registration in the presence of large motions. Pravin Bhat, Ke Colin Zheng, Noah Snavely, Aseem Agarwala, Maneesh Agrawala, Michael Cohen, Brian Curless, CVPR 2006, New York, NY, June 2006.
- A Theory of Spherical Harmonic Identities for BRDF/Lighting Transfer and Image Consistency . Dhruv Mahajan, Ravi Ramamoorthi and Brian Curless, ECCV 2006, Graz, Austria, May 2006.
- Shape and Spatially-Varying BRDFs From Photometric Stereo. Dan B. Goldman, Brian Curless, Aaron Hertzmann, Steve Seitz, ICCV 2005, Beijing, China, November, 2005.
- Animating pictures with stochastic motion textures. Yung-Yu Chuang, Dan B. Goldman, Colin Zheng, Brian Curless, David Salesin, and Rick Szeliski. SIGGRAPH 2005, Los Angeles, CA, August 2005.
- Panoramic video textures. Aseem Agarwala, Colin Zheng, Chris Pal, Maneesh Agrawala, Michael Cohen, Brian Curless, David Salesin, and Rick Szeliski. SIGGRAPH 2005, Los Angeles, CA, August 2005.
- Spacetime faces: High resolution capture for modeling and animation. Li Zhang, Noah Snavely, Brian Curless, and Steve Seitz. SIGGRAPH 2004, Los Angeles, CA, August 2004.
- Interactive digital photomontage. Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, Michael Cohen. SIGGRAPH 2004, Los Angeles, CA, August 2004.
- Shape and motion under varying illumination: unifying structure from motion, photometric stereo, and multi-view stereo. Li Zhang, Brian Curless, and Steve Seitz. ICCV 2003, Nice, France, October, 2003.

- The space of human body shapes: reconstruction and parameterization from range scans. Brett Allen, Brian Curless, and Zoran Popović. SIGGRAPH 2003, San Diego, CA, July, 2003.
- Shadow matting and compositing. Yung-Yu Chuang, Dan Goldman, Brian Curless, David Salesin, and Rick Szeliski. SIGGRAPH 2003, San Diego, CA, July, 2003.
- Articulated Body Deformation from Range Scan Data. Brett Allen, Brian Curless, and Zoran Popović, ACM Transactions on Graphics, SIGGRAPH 2002, San Antonio, Texas, pp. 612-619, July 2002.
- Interactive Skeleton-Driven Dynamic Deformations. Steve Capell, Seth Green, Brian Curless, Tom Duchamp, Zoran Popović. ACM Transactions on Graphics, SIGGRAPH 2002, San Antonio, Texas, pp. 586-593, July 2002.
- Video matting of complex scenes. Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David Salesin, and Richard Szeliski. ACM Transactions on Graphics, SIGGRAPH 2002, San Antonio, Texas, pp. 243-248, July 2002.
- A multiresolution framework for dynamic deformations. Steve Capell, Seth Green, Brian Curless, Tom Duchamp, Zoran Popović. ACM SIGGRAPH Symposium on Computer Animation (SCA 2002), San Antonio, Texas, pp. 41-47, July 2002.
- Curve Analogies. Aaron Hertzmann, Nuria Oliver, Brian Curless, and Steve Seitz. Eurographics Workshop on Rendering (EGWR) 2002, Pisa, Italy, pp. 233-245, June 2002.
- Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. Li Zhang, Brian Curless, and Steve Seitz. IEEE International Symposium on 3D Data Processing Visualization and Transmission (3DPVT 2002), Padova, Italy, pp. 24-36, June 2002.

Scientific and professional societies of which a member:

- Association for Computing Machinery
- Institute of Electrical and Electronics Engineers, Inc.

Honors and awards:

- UW ACM Teaching Award, University of Washington (2004)
- Sloan Fellowship for Computer Science, University of Washington (2000)
- NSF CAREER Award, University of Washington (1999)

Institutional and professional service in last five years:

- Co-editor-in-chief, Foundations and Trends in Computer Graphics and Computer Vision (2003 - present).
- Co-organized a "3D Photography" course for CVPR'99, SIGGRAPH'99, SIGGRAPH '00
- Presenter for "Digital geometry processing" course submitted (and accepted) for SIGGRAPH '01
- Reviewer for SIGGRAPH '00
- Reviewer for IEEE Transactions on Visualization and Computer Graphics
- Reviewer for Computers and Graphics

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - medium.

Diorio, Chris

Assistant professor (1997 Ass't)

Education:

- B.A., Physics, Occidental College, 1993
- M.S., Electrical Engineering, California Institute of Technology, 1984
- Ph.D., Electrical Engineering, California Institute of Technology, 1997

Other Experience:

- Senior Staff Engineer, TRW, Inc., 1991-1997
- Senior Staff Scientist, American Systems Corporation, 1989-1991
- Technical Consultant, American Systems Corporation, 1988-1989
- Customer Technical Representative, The Analytic Sciences Corporation, 1986-1989
- Member of the Technical Staff, TRW, Inc., 1984-1986

Consulting, patents, etc.:

- C. Diorio and C. Mead, "pMOS EEPROM nonvolatile data storage," U.S. Patent No. 6,144,581, issued 7 Novem-ber, 2000.
- C. Diorio and C. Mead, "Semiconductor structures for long-term learning," U.S. Patent No. 6,125,053, issued 26 September, 2000.
- C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "Hole impact-ionization method of hot-electron injection and four-terminal pFET semiconductor structure for long-term learning," U.S. Patent No. 5,990,512, issued 23 Novem-ber, 1999.
- C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "Method for implementing a learning function," U.S. Patent No. 5,914,894, issued 22 June, 1999.
- C. Diorio and C. Mead, "A pMOS analog EEPROM cell," U.S. Patent No. 5,898,613, issued 27 April, 1999.
- B. A. Minch, P. Hasler, C. Diorio, and C. Mead, "An autozeroing floating-gate amplifier," U.S. Patent No. 5,986,927, issued 16 November, 1999.

Principal publications of last five years:

- Refereed Journal Publications
- A. Pesavento, T. Horiuchi, C. Diorio, and C. Koch, "Adaptation of current signals with floating-gate circuits," *Analog Integrated Circuits and Signal Processing*, in press.
- M. Figueroa, D. Hsu, and C. Diorio, "A mixed-signal approach to high-performance, low-power linear filters," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 816-822, 2001.
- D. Hsu, M. Figueroa, and C. Diorio, "A silicon primitive for competitive learning," *IEEE Trans. Neural Networks*, in press.
- P. Hasler, B. A. Minch, and C. Diorio, "An autozeroing floating-gate amplifier," *IEEE Trans. Circuits and Systems II*, vol. 48, no. 1, pp. 74-82, 2001.
- B. A. Minch, P. Hasler, and C. Diorio, "Multiple-input translinear element networks," *IEEE Trans. Circuits and Systems II*, vol. 48, no. 1, pp. 20-28, 2001.
- C. Diorio, "A p-channel MOS synapse transistor with self-convergent memory writes," *IEEE Trans. Electron De-vices*, vol. 47, no. 2, pp. 464-472, 2000.
- C. Diorio, T. Humes, H. Notthoff, G. Chao, A. Lai, J. Hyde, M. Kintis, and A. Oki, "A low-noise, GaAs/AlGaAs, microwave frequency-synthesizer IC," *IEEE J. Solid-State Circuits*, vol. 33, no. 9, pp. 1306-1312, 1998.
- P. Hasler, A. G. Andreou, C. Diorio, B. A. Minch, and C. Mead, "Impact ionization and hot-electron

injection de-rived consistently from Boltzmann transport," VLSI Design, vol. 8, no. 1-4, pp. 455-461, 1998.

- C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "A floating-gate MOS learning array with locally computed weight updates," IEEE Trans. Electron Devices, vol. 44, no. 12, pp. 2281-2289, 1997.
- C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "A complementary pair of four-terminal silicon synapses," Analog Integrated Circuits and Signal Processing, vol. 13, no. 1/2, pp. 153-166, 1997.
- C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "A single-transistor silicon synapse," IEEE Trans. Electron De-vices, vol. 43, no. 11, pp. 1972-1980, 1996.
- B. A. Minch, C. Diorio, P. Hasler, and C. Mead, "Translinear circuits using subthreshold floating-gate MOS tran-sistors," Analog Integrated Circuits and Signal Processing, vol. 9, no. 2, pp. 167-179, 1996.
- Refereed Conference Publications
- D. Hsu, M. Figueroa, and C. Diorio, "A silicon primitive for competitive learning," Advances in Neural Informa-tion Processing Systems 13, MIT Press, Cambridge, MA, 2001 (NIPS conference held Nov. 27-30 2000; proceed-ings in press).

Scientific and professional societies of which a member:

- Member, IEEE

Honors and awards:

- A UW Distinguished Teaching Award on June 7, 2001.
- An ONR Young Investigator Award on February 6, 2001.
- An Alfred P. Sloan Foundation Research Fellowship on Feb. 4, 2000.
- Distinguished Lecture Series invitee, the University of Virginia, Department of Computer Science, Jan. 24, 2000. Talk title: "Biologically Inspired Computation".
- An NSF Presidential Early Career Award in Science and Engineering (PECASE) on Feb. 10, 1999.
- A five-year Packard Foundation Fellowship in Science and Engineering from the David and Lucile Packard Foundation in September, 1998.
- An NSF CAREER Award on May 15, 1998.
- The Electron Devices Society's Paul Rappaport Award for the best paper in an IEEE EDS publication in 1996 for "A single-transistor silicon synapse," IEEE Trans. Electron Devices, vol. 43, no. 11, pp. 1972-1980, 1996.

Institutional and professional service in last five years:

- 2000-2001 Faculty recruiting, staff ombudsperson
- 1999-2000 Faculty recruiting, staff ombudsperson, Endowed Chair selection, undergraduate program expansion
- 1998-1999 Adjunct to the EE faculty recruiting committee
- 1997-1998 CSE co-op liaison
- Sixteen press interviews and public-interest stories, including radio and television spots, quotes in Scientific American, etc. See <http://www.cs.washington.edu/people/faculty/diorio/Talks/index.html>.

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - medium.

Domingos, Pedro

Associate professor (1999 Ass't, 2004 Assoc)

Education:

- Licenciatura, Electrical Engineering and Computer Science, Instituto Superior Tecnico, Lisbon, Portugal, 1988
- M.S., Electrical Engineering and Computer Science, Instituto Superior Tecnico, Lisbon, Portugal, 1992
- M.S., Information and Computer Science, University of California at Irvine, 1994
- Ph.D., Information and Computer Science, University of California at Irvine, 1997

Other Experience:

- Assistant professor, Instituto Superior Tecnico, Portugal (1997-99)
- Research and teaching assistant, Instituto Superior Tecnico, Portugal (1987-92)
- Intern, INESC, Portugal (1986-88)
- Researcher, INESC, Portugal (1988-89)

Consulting, patents, etc.:

- Irvine Research Corporation (1994)

Principal publications of last five years:

- P. Domingos, C. Faloutsos, T. Senator, H. Kargupta and L. Getoor (editors), Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 2003.
- M. Richardson and P. Domingos, "Combining Link and Content Information in Web Search," in Web Dynamics (pp. 179-193), edited by M. Levene and A. Poullovassilis, Springer, 2004.
- A. Doan, J. Madhavan, P. Domingos and A. Halevy, "Ontology Matching: A Machine Learning Approach," in Handbook on Ontologies in Information Systems (pp. 385-403), edited by S. Staab and R. Studer, Springer, 2004.
- P. Domingos, "Machine Learning," in Handbook of Data Mining and Knowledge Discovery (pp. 660-670), edited by W. Klossgen and J. Zytkow, Oxford University Press, 2002.
- F. Provost and P. Domingos, "Tree Induction for Probability-Based Ranking," Machine Learning, vol. 52 (pp. 199-216), 2003.
- A. Doan, P. Domingos and A. Halevy, "Learning to Match the Schemas of Data Sources: A Multistrategy Approach," Machine Learning, vol. 50 (pp. 279-301), 2003.
- D. Lowd and P. Domingos, "Recursive Random Fields," in Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, Hyderabad, India, 2007.
- P. Singla and P. Domingos, "Entity Resolution with Markov Logic," in Proceedings of the Sixth IEEE International Conference on Data Mining (pp. 572-582), Hong Kong, 2006.
- H. Poon and P. Domingos, "Sound and Efficient Inference with Probabilistic and Deterministic Dependencies," in Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 458-463), Boston, MA, 2006.
- S. Kok and P. Domingos, "Learning the Structure of Markov Logic Networks," in Proceedings of the Twenty-Second International Conference on Machine Learning (pp. 441-448), Bonn, Germany, 2005.
- D. Grossman and P. Domingos, "Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood," in Proceedings of the Twenty-First International Conference on Machine Learning (pp. 361-368), Banff, Canada, 2004.
- R. Dhamankar, Y. Lee, A. Doan, A. Halevy and P. Domingos, "iMAP: Discovering Complex Semantic Matches between Database Schemas," in Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data (pp. 383-394), Paris, France, 2004.

- M. Richardson and P. Domingos, "Learning with Knowledge from Multiple Experts," in Proceedings of the Twentieth International Conference on Machine Learning (pp. 624-631), Washington, DC, 2003.
- M. Richardson and P. Domingos, "Building Large Knowledge Bases by Mass Collaboration," in Proceedings of the Second International Conference on Knowledge Capture (pp. 129-137), Sanibel Island, FL, 2003.
- T. Lau, P. Domingos and D. Weld, "Learning Programs from Traces Using Version Space Algebra," in Proceedings of the Second International Conference on Knowledge Capture (pp. 36-43), Sanibel Island, FL, 2003.
- M. Richardson, R. Agrawal and P. Domingos, "Trust Management for the Semantic Web," in Proceedings of the Second International Semantic Web Conference (pp. 351-368), Sanibel Island, FL, 2003.
- Conference on Artificial Intelligence (pp. 80-86), Edmonton, Canada, 2002.
- A. Doan, J. Madhavan, P. Domingos and A. Halevy, "Learning to Map between Ontologies on the Semantic Web," in Proceedings of the Eleventh International World Wide Web Conference (pp. 662-673), Honolulu, HI, 2002.

Scientific and professional societies of which a member:

- ACM / SIGKDD, SIGART, SIGMOD
- IEEE / Computer Society
- AAAI

Honors and awards:

- NSF CAREER Award (2000)
- IBM Faculty Partnership Award (2000)
- KDD-99 Best Paper Award
- KDD-98 Best Paper Award
- Fulbright Scholarship (1992-97)
- University of California Regents' Dissertation Fellowship (1996)

Institutional and professional service in last five years:

- Editorial board member, Machine Learning (current)
- Editorial board member, Journal of Artificial Intelligence Research (current)
- Editorial board member, Intelligent Data Analysis (current)
- Editorial board member, Applied Intelligence (current)
- Editorial board member, Evaluation of Intelligent Systems (current)
- Area chair (supervised learning), ICML-2001
- Panels chair, KDD-2001
- Program committee member, AAAI-97, AAAI-98, AAAI-99, AAAI-2000
- Program committee member, ICML-98, ICML-99, ICML-2000
- Program committee member, KDD-98, KDD-99

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Ebeling, Carl

Professor (1986 Ass't, 1992 Assoc, 1997 Full)

Education:

- B.S., Physics, Wheaton College, 1971
- M.S., Computer Science, Southern Illinois University - Carbondale, 1976
- Ph.D., Computer Science, Carnegie-Mellon University, 1986

Other Experience:

- Quiksilver Technology: Visting Fellow 2000-2001

Consulting, patents, etc.:

- Carl Ebeling, Gaetano Borriello, Scott Hauck and Steve Burns. "A Dynamically Reconfigurable Logic Array for Digital Logic Circuits," U. S. Patent #5,208,491. 1993.
- Scott Hauck, Gaetano Borriello, Steve Burns, and Carl Ebeling. "A Field-Programmable Gate Array for Synchronous and Asynchronous Operation," U. S. Patent #5,367,209. 1994.
- Carl Ebeling, Darren Cronquist and Paul Franklin. "Reconfigurable computing architecture for providing pipelined data paths", U.S. Patent #6,023,742. 2000.
- Carl Ebeling, Eugene Hogenauer. "Method, system and software for programming reconfigurable hardware", U.S. Patent # 6,732,354, 2004.

Principal publications of last five years:

- Akshay Sharma, Carl Ebeling, and Scott Hauck. PipeRoute: A Pipelining-Aware Router for FPGAs, In Proceedings of the International Symposium on Field-Programmable Gate Arrays, Monterey, pp. 68-77, February, 2003.
- Carl Ebeling, Chris Fisher, Guanbin Xing, Manyuan Shen and Hui Liu. "Implementing an OFDM Receiver on the RaPiD Reconfigurable Architecture", Proceedings of the 13th International Conference on Field-Programmable Logic and Applications (FPL 2003), Lisbon, pp. 21-30, August 2003.
- John F. Keane, Christopher Bradley, and Carl Ebeling. "A Compiled Accelerator for Biological Cell Signalling Simulations," Twelfth International Symposium on Field-Programmable Gate Arrays (FPGA'04), pp 233-242, Monterey, CA, February 2004.
- Akshay Sharma, Katherine Compton, Carl Ebeling and Scott Hauck. "Exploration of Pipelined FPGA Interconnect Structures," Twelfth International Symposium on Field-Programmable Gate Arrays (FPGA'04), pp. 13-22, Monterey, CA, February 2004.
- Carl Ebeling, Chris Fisher, Guanbin Xing, Manyuan Shen and Hui Liu. "Implementing an OFDM Receiver on the RaPiD Reconfigurable Architecture," IEEE Transactions on Computers, Vol. 53, Issue 11, pp.1436-1448, Nov. 2004.
- Song Li and Carl Ebeling. "QuickRoute: A Fast Routing Algorithm for Pipelined Architectures", IEEE International Conference on Field-Programmable Technology, pp. 73-80, Dec. 2004. (Best student paper award)
- Akshay Sharma, Scott Hauck and Carl Ebeling. "Architecture Adaptive Routability-Driven Placement for FPGAs," Proceedings of the International Conference on Field-Programmable Logic and Applications (FPL 2005), pp. 427-432, Sept. 2005.
- Akshay Sharma, Carl Ebeling and Scott Hauck. "PipeRoute: A Pipelining-Aware Router for Reconfigurable Architectures," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 25, pp. 518-532, March 2006.
- Benjamin Ylvisaker, Brian Van Essen and Carl Ebeling. "A Type Architecture for Hybrid Micro-Parallel Computers," Proceedings of the International Symposium on Field-Programmable Custom Computing

Machines (FCCM), April 2006.

- Allan Carroll and Carl Ebeling. "Reducing the Space Complexity of Pipelined Routing using Modified Range Encoding," Proceedings of the International Conference on Field-Programmable Logic and Applications (FPL 2006), Sept. 2006. (Best paper on FPGA technology)

Scientific and professional societies of which a member:

- Association for Computing Machinery
- IEEE
- Sigma Xi

Honors and awards:

- Association for Computing Machinery Distinguished Dissertation Award 1986
- National Science Foundation Presidential Young Investigator 1987
- American Association for Artificial Intelligence Pioneer in Computer Chess 1989
- Burlington Resources Foundation Faculty Achievement Award for Teaching, College of Engineering 1992
- Fulbright Fellowship, University of Mauritius 1993-94
- University of Washington Distinguished Teaching Award 1995
- Allen Newell Award for Research Excellence 1997

Institutional and professional service in last five years:

- Steering Committee, ACM Symposium on Field-Programmable Gate Arrays, 1997-2006.
- Program Committee, ACM Symposium on Field-Programmable Gate Arrays, 2006-2007.
- Program Committee, International Conference on Field Programmable Logic and Applications, 1997-2007.
- Steering Committee, Reconfigurable Architectures Workshop, 2004
- Steering Committee and Program Committee, Conference on the Engineering of Reconfigurable Systems and Algorithms (ERSA), 2004
- Undergraduate Program Coordinator, 2005-2007
- College of Engineering Accreditation and Continuous Improvement Committee, 2005-2007

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - medium.

Eggers, Susan

Professor (1989 Ass't, 1994 Assoc, 1999 Full)

Education:

- B.A., Economics, Connecticut College, 1965
- M.S., Computer Science, University of California, Berkeley, 1984
- Ph.D., Computer Science, University of California, Berkeley, 1989

Other Experience:

- 1979 - 1983: Computer Scientist, Lawrence Berkeley Laboratory,
- Department of Computer Science and Mathematics.

Consulting, patents, etc.:

- 1997: Consultant, Compaq (then Digital) Western Research Laboratory.
- Pipelined Multi-thread Processor Selecting Thread Instruction in Inter-stage Buffer Based on Count Information: No. 6,470,443, 10/22/02. Licensed to Intel Corp.
- System and Method for Performing Selective Dynamic Compilation Using Run-time Information: No. 6,427,234 B1, 7/30/02.
- Register-Freeing Mechanism for Dynamic Out-of-Order Processors Using Register Renaming: No. 6,314,511 B2, 11/6/01. Licensed to Intel Corp.
- Shared Register Storage Mechanisms for Multithreading Computer Systems with Out-of-Order Execution, No.: 6,092,175, 7/18/00. Licensed to Intel Corp.
- Thread Properties Attribute Vector Based Thread Selection in Multithreading Processor, No.: 6,073,159, 6/6/00. Licensed to Intel Corp.

Principal publications of last five years:

- S. Swanson, A. Schwerin, M. Mercaldi, A. Petersen, A. Putnam, K. Michelson, M. Oskin, and S.J. Eggers, "The WaveScalar Architecture", to appear in Transactions on Computer Systems.
- A. Petersen, M. Mercaldi, A. Putnam, A. Schwerin, S. Swanson, M. Oskin, and S.J. Eggers, "Controlling Control Overhead in Distributed Dataflow Architectures", Parallel Architectures and Compilation Techniques (September, 2006), pp. 182-191.
- M. Mercaldi, A. Petersen, A. Putnam, A. Schwerin, S. Swanson, M. Oskin, and S.J. Eggers, "Modeling Instruction Placement on a Spatial Architecture", Symposium on Parallel Algorithms and Architectures (July, 2006), pp. 158-169.
- M. Mercaldi, S. Swanson, A. Petersen, A. Putnam, A. Schwerin, M. Oskin, and S.J. Eggers, "Instruction Scheduling for a Tiled Architecture", International Conference on Architectural Support for Programming Languages and Operating Systems (July, 2006), pp. 141-150.
- L.K. McDowell, S.J. Eggers S.D. Gribble, "Improving Server Software Support for Simultaneous Multithreaded Processors", Symposium on Principles and Practice of Parallel Programming (June, 2003)
- J. Redstone, S.J. Eggers and H.M. Levy, "Mini-threads: Increasing TLP on Small-Scale SMT Processors," International Conference on High-Performance Computer Architecture (February 2003), pp. 19-30.94), pp. 176-186.
- J. Aldrich, E.G. Sirer, C. Chambers and S.J. Eggers, "Comprehensive Synchronization Elimination for Java", Science of Computer Programming, 47(2-3):91-120, (May-June 2003).
- M. Philipose, C. Chambers, and S.J. Eggers. Towards Automatic Construction of Staged Compilers. Symposium on Principles of Programming Languages (January 2002), pp. 113-125.

Scientific and professional societies of which a member:

- ACM
- IEEE
- AAAS
- NAE

Honors and awards:

- 2006: Fellow, American Association for the Advancement of Science
- 2006: Member, National Academy of Engineering
- 2004: Microsoft Professor of Computer Science & Engineering
- 2002: Fellow, Institute of Electrical & Electronics Engineers
- 2002: Fellow, Association of Computing Machinery
- 2000 - 2001: IBM Faculty Partnership Award
- 1993: Finalist, NSF Presidential Faculty Fellowship
- 1991 - 1994: AT&T Bell Laboratories, PYI Matching Program
- 1991: Finalist, David and Lucile Packard Foundation Fellowship
- 1990 - 1995: NSF Presidential Young Investigator Award
- 1989 - 1991: IBM Faculty Development Award
- 1986 - 1988: IBM Graduate Student Fellowship
- 1985 - 1986: California Fellowship in Microelectronics

Institutional and professional service in last five years:

- General Chair, International Conference on Architectural Support for Programming Languages and Operating Systems, 2008.
- Member, NSF Review Panel on Computing Processes and Artifacts, 2006.
- Member, University College/School Organization Committee, 2006.
- Member, Computer Science & Engineering Chair Search Committee, 2006
- Chair, College of Engineering Promotion & Tenure Committee, 2005/06
- Member, College of Engineering Promotion & Tenure Committee, 2006/07, 2004/05, 2001/03
- Associate Chair, 2002/03
- Program Committee, International Symposium on Computer Architecture, 2006, 2005, 2003, 2002. Member,
- Program Committee, NSF Workshop on Grand Challenges in Computer Architecture, 2005.
- Program Committee, International Symposium on Code Generation & Optimization, 2003.
- ACM Eckert-Mauchly Award Committee, 2002-2005.
- Invited Speaker, CRA Workshop on Academic Careers in Computer Science, 2002.
- CISE Advisory Board, National Science Foundation, 2001-2004.
- Steering Committee, International Symposium on Computer Architecture, 2000-2002.
- Civil & Environmental Engineering Chair Search Committee, 2001/02

States in which registered: ?

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Etzioni, Oren

Professor (1991 Ass't, 1996 Assoc, 2005 Full)

Education:

- Ph.D., Computer Science, Carnegie Mellon University, 1991
- M.S., Computer Science, Carnegie Mellon University, 1988
- B.A., Computer Science, Harvard University, 1986

Other Experience:

- Director, UW Turing Center, www.turing.washington.edu, May 2005 - Present
- Chief Technology Officer, Go2Net (acquired by Infospace in 2000), 1999-2000
- Founder and Chief Scientist, Netbot (acquired by Excite in 1997), 1996-1997
- Venture Partner at Madrona Venture Group, 2000–Present

Consulting, patents, etc.:

- Founder, Farecast (formerly known as Hamlet), 2004
- Founder, ClearForest (acquired by Reuters in 2007), 2000
- Board Member, Performant (acquired by Mercury Interactive in 2003)

Principal publications of last five years:

- Doug Downey, Stefan Schoenmackers, and Oren Etzioni. Sparse Information Extraction: Unsupervised Language Models to the Rescue. Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007).
- Alexander Yates and Oren Etzioni. Unsupervised Resolution of Objects and Relations on the Web. Proceedings of Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007).
- Oren Etzioni, Michele Banko, and Michael J. Cafarella. Machine Reading. Proceedings of the 2007 AAAI Spring Symposium on Machine Reading.
- Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open Information Extraction from the Web. Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007).
- Doug Downey, Matthew Broadhead, and Oren Etzioni. Locating Complex Named Entities in Web Text. Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007).
- Oren Etzioni, Michele Banko, and Michael J. Cafarella. Machine Reading. Proceedings of the 21th National Conference on Artificial Intelligence (AAAI 2006).
- Marcus Sammer, Kobi Reiter, Stephen Soderland, Katrin Kirchhoff, and Oren Etzioni. Ambiguity Reduction for Machine Translation: Human-Computer Collaboration. Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA 2006).
- Alexander Yates, Stefan Schoenmackers, and Oren Etzioni. Detecting Parser Errors Using Web-based Semantic Filters. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006).
- Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Methods for domain-independent information extraction from the Web: an experimental comparison. Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004).
- Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in KnowItAll (preliminary results). Proceedings of the 13th International Conference on the World Wide Web (WWW

2004).

- Luke McDowell , Oren Etzioni, Alon Halevy, and Henry Levy. Semantic Email. Proceedings of the 13th International Conference on the World Wide Web (WWW 2004). Runner-up for Best Paper Award.
- Oren Etzioni, Alon Halevy, Henry Levy, and Luke McDowell. Semantic Email: Adding Lightweight Data Manipulation Capabilities to the Email Habitat. Proceedings of the 6th International Workshop on the Web and Databases (WebDB 2003).
- Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. Towards a theory of natural language interfaces to databases. Proceedings of the Intelligent User Interfaces (IUI 2003).
- Alexander Yates, Oren Etzioni, and Daniel S. Weld. A reliable natural language interface to household appliances. Proceedings of the International Conference on Intelligent User Interfaces (IUI 2003).

Scientific and professional societies of which a member:

- Association for the Advancement of Artificial Intelligence
- Association for Computing Machinery
- Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining
- Association for Computational Linguistics

Honors and awards:

- 2003: Association for the Advancement of Artificial Intelligence Fellow
- 1997: Edge Award for the best use of intelligent technology (in Jango) at the WebINNOVATION show.
- 1997: MetaCrawler selected for Editor's Choice Award for Web Search Engines by PC Magazine (December).
- 1997: MetaCrawler selected as Best Web Search Solution by Infoworld (May).
- 1993: NSF Young Investigator Award.
- 1992: NSF Research Initiation Award.
- 1987-1990: AT&T Bell Laboratories Fellow.

Institutional and professional service in last five years:

- 2004–2005: Faculty Recruiting Committee Member, Computer Science & Engineering.
- 2003–2004: Graduate Program Coordinator Chair, Computer Science & Engineering.
- 2002–2003: Graduate Program Coordinator Chair, Computer Science & Engineering.
- 2002–2003: Graduate Admissions Committee Member, Computer Science & Engineering.
- 2002–2003: Executive Committee Member, Computer Science & Engineering.
- Councilor, Association for the Advancement of Artificial Intelligence, 2004–2007.
- President, AI Access Foundation.
- Associate Editor, ACM Transactions on the Web.
- Journal of Computational Intelligence
- Journal of Data Mining and Knowledge Discovery

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - high.

Fox, Dieter

Assistant professor (2000 Ass't, 2005 Assoc)

Education:

- B.S., Computer Science, University of Bonn, Germany, 1990
- M.S., Computer Science, University of Bonn, Germany, 1993
- Ph.D., Computer Science, University of Bonn, Germany, 1998

Other Experience:

- 8/98 - 8/00 Postdoctoral research associate, Department of Computer Science, Carnegie Mellon University

Consulting, patents, etc.:

- Intel Corporation

Principal publications of last five years:

- G. Sukhatme, S. Schaal, W. Burgard, and D. Fox, editors. Robotics Science and Systems. MIT Press, 2007.
- S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics. MIT Press, Cambridge, MA, September 2005. ISBN 0-262-20162-3.
- M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. H. Ahnel, D. Fox, and H. Kautz. Inferring activities from interactions with objects. IEEE Pervasive Computing Magazine, 3(4):50-57, 2004.
- C.T. Kwok, D. Fox, and M. Meil \square a. Real-time particle filters. Proceedings of the IEEE, 92(2):469-484, 2004. Special Issue on Sequential State Estimation.
- D. Fox, J. Ko, K. Konolige, and B. Stewart. A hierarchical Bayesian approach to mobile robot map structure learning. In P. Dario, R. Chatila, editors, Robotics Research: the Eleventh International Symposium, Springer Tracts in Advanced Robotics (STAR). Springer Verlag, 2004.
- K. Konolige, C. Ortiz, R. Vincent, A. Agno, M. Eriksen, B. Limetkai, M. Lewis, L. Briesemeister, E. Ruspini, D. Fox, J. Ko, B. Stewart, and L. Guibas. CentiBOTS: Large-scale robot teams. In A. Schultz, L. Parker, and F. Schneider, editors, Multi-Robot Systems: From Swarms to Intelligent Automata, volume II, pages 193–204. Kluwer, 2003.
- C. Plagemann, D. Fox, and W. Burgard. Efficient failure detection on mobile robots using particle filters with Gaussian process proposals. In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI), 2007.
- B. Ferris, D. Fox, and N. Lawrence. WiFi-SLAM using gaussian process latent variable models. In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI), 2007.
- L. Liao, D. Fox, and H. Kautz. Location-based activity recognition. In Advances in Neural Information Processing Systems (NIPS), 2005.
- J. Letchner, D. Fox, and A. LaMarca. Large-scale localization from wireless signal strength. In Proc. of the National Conference on Artificial Intelligence (AAAI), 2005.
- L. Liao, D. Fox, and H. Kautz. Location-based activity recognition using relational Markov networks. In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI), 2005.
- D. Schulz, D. Fox, and J. Hightower. People tracking with anonymous and id-sensors using Rao-Blackwellised particle filters. In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI), pages 921–926, 2003.
- J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3232–3237, 2003.
- A. LaMarca, W. Brunette, D. Koizumi, M. Lease, S.B. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello. Making sensor networks practical with robots. In International Conference on Pervasive Computing

(Pervasive), 2002.

- A. LaMarca, W. Brunette, D. Koizumi, M. Lease, S.B. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello. PlantCare: An investigation in practical ubiquitous systems. In International Conference on Ubiquitous Computing (UbiComp), 2002.

Scientific and professional societies of which a member:

- Institute of Electrical and Electronics Engineers
- American Association for Artificial Intelligence
- Gesellschaft fuer Informatik

Honors and awards:

- Best Paper Award: International Symposium on Wearable Computers (ISWC), 2005.
- Outstanding paper award: National Conference on Artificial Intelligence (AAAI), 2004.
- Scientific Challenge Award: Robocup Symposium 2004.
- NSF CAREER Award: Probabilistic Methods for Multi-Robot Collaboration, 2001.
- ECCAI Artificial Intelligence Dissertation Award: European Coordinating Committee for Artificial Intelligence, 2000.
- Best paper award: IEEE International Conference on Robotics & Automation (ICRA), 2000.
- AKI Dissertation Award: Arbeitsgemeinschaft der deutschen KI-Institute (German AI institutes), 1999.

Institutional and professional service in last five years:

- Program co-chair Twenty-Third Conference on Artificial Intelligence (AAAI), 2008.
- Associate Editor IEEE Transactions on Robotics (T-RO), 2004 – 2007.
- Editorial Board Journal of Artificial Intelligence Research (JAIR), 2003 – 2006.
- Co-chair NIPS-05 Workshop on Activity Recognition and Discovery, Neural Information Processing Systems (NIPS), 2005.
- Publication chair IEEE International Conference on Advanced Robotics (ICAR), 2005.
- Co-chair NIPS-04 Workshop on Activity Recognition and Discovery, Neural Information Processing Systems (NIPS), 2004.
- Neural Information Processing Systems (NIPS), 2006 (area chair).
- National Conference on Artificial Intelligence (AAAI), 1998, 1999, 2000, 2002 (senior PC).
- International Joint Conference on Artificial Intelligence (IJCAI), 2003.
- Robocup Symposium, 2003.
- German Conference on Artificial Intelligence (KI), 2002.
- Starting Artificial Intelligence Researchers Symposium (STAIRS), 2002.
- KI Workshop on Uncertainty in Robotics and Multi-Agent Systems (KI), 2002.
- Workshop on Interactive Robotics and Entertainment (WIRE), 2000.
- IEEE International Symposium on Computational Intelligence in Robotics & Automation, 1999.

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Fogarty, James

Assistant professor (2006 Ass't)

Education:

- Ph.D., Human-Computer Interaction, Carnegie Mellon University, 2006
- B.S., Computer Science, Virginia Tech, 2000

Principal publications of last five years:

- Fogarty, J., Hudson, S.E., Atkeson, C.G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J.C., and Yang, J. (2005). Predicting Human Interruptibility with Sensors. *ACM Transactions on Computer Human Interaction (TOCHI)*, Vol. 12, No. 1, March 2005, pp. 119-146.
- Fogarty, J., Lai, J., and Christensen, J. (2004). Presence versus Availability: The Design and Evaluation of a Context Aware Communication Client. *International Journal of Human Computer Studies (IJHCS)*, Vol. 61, No. 3, September 2004, pp. 299-317.
- Fogarty, J. and Hudson, S.E. (2007). Toolkit Support for Developing and Deploying Sensor Based Statistical Models of Human Situations. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2007)*, to Appear. (Acceptance Rate: 24%)
- Avrahami, D., Fogarty, J., and Hudson, S.E. (2007). Biases in Human Estimation of Interruptibility: Effects and Implications for Practice. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2007)*, to Appear. (Acceptance Rate: 24%)
- Tullio, J., Dey, A.K., Chalecki, J., and Fogarty, J. (2007). How it Works: A Field Study of Non Technical Users Interacting with an Intelligent System. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2007)*, to Appear. (Acceptance Rate: 24%)
- Andrew, A., Borriello, G., and Fogarty, J. (2007). Toward a Systematic Understanding of Suggestion Tactics in Persuasion Technologies. *Proceedings of the Conference on Persuasive Technology (Persuasive 2007)*, to Appear.
- Fogarty, J., Au, C., and Hudson, S.E. (2006). Sensing from the Basement: A Feasibility Study of Unobtrusive and Low-Cost Home Activity Recognition. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2006)*, pp. 91 100. (Acceptance Rate: 22%)
- Tang, K.P., Keyani, P., Fogarty, J., and Hong, J.I. (2006). Putting People in their Place: An Anonymous and Privacy Sensitive Approach to Collecting Sensed Data in Location Based Applications. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2006)*, pp. 93 102. (Acceptance Rate: 23%)
- Fogarty, J., Baker, R.S., and Hudson, S.E. (2005). Case Studies in the use of ROC Curve Analysis for Sensor Based Estimates in Human Computer Interaction. *Proceedings of Graphics Interface (GI 2005)*, pp. 129 136. (Acceptance Rate: 28%)
- Fogarty, J., Ko, A.J., Aung, H.H., Golden, E., Tang, K.P., and Hudson, S.E. (2005). Examining Task Engagement in Sensor Based Statistical Models of Human Interruptibility. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2005)*, pp. 331-340. (Acceptance Rate: 25%)
- Fogarty, J., Hudson, S.E., and Lai, J. (2004). Examining the Robustness of Sensor Based Statistical Models of Human Interruptibility. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2004)*, pp. 207-214. (Acceptance Rate: 16%)
- Fogarty, J. and Hudson, S.E. (2003). GADGET: A Toolkit for Optimization Based Approaches to Interface and Display Generation. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2003)*, pp. 125 134. (Acceptance Rate: 21%)
- Fogarty, J., Forlizzi, J., and Hudson, S.E. (2003). Portrait: Generating Personal Presentations. *Proceedings of Graphics Interface (GI 2003)*, pp. 209 216. (Acceptance Rate: 33%)
- Hudson, S.E., Fogarty, J., Atkeson, C.G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J.C., and Yang, Y. (2003). Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2003)*, pp. 257-264. (Acceptance Rate: 16%)

- Fogarty, J., Forlizzi, J., and Hudson, S.E. (2002). Specifying Behavior and Semantic Meaning in an Unmodified Layered Drawing Package. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2002), pp. 61 70. (Acceptance Rate: 22%)

Institutional and professional service in last five years:

- Program Committee member
- UbiComp 2007: International Conference on Ubiquitous Computing
- Program Committee member
- GI 2007: Graphics Interface
- Program Committee member
- UIST 2006: ACM Symposium on User Interface Software and Technology
- Program Committee member
- GI 2006: Graphics Interface
- Poster Program Committee member
- UIST 2005: ACM Symposium on User Interface Software and Technology
- Workshop Program Committee member
- UbiComp 2005: International Conference on Ubiquitous Computing
- Student Volunteer Co Chair (with Joe Tullio)
- UIST 2004: ACM Symposium on User Interface Software and Technology
- IHCS: International Journal of Human Computer Studies (2005)
- TOCHI: ACM Transactions on Computer Human Interaction (2003)
- CHI: ACM Conference on Human Factors in Computing Systems (2003-2007)
- CSCW: ACM Conference on Computer Supported Cooperative Work (2004, 2006)
- Eurographics: Conference of the European Association for Computer Graphics (2005)
- GI: Graphics Interface (2005 2007)
- HCI: Human-Computer Interaction (2006)
- Intelligent Systems: IEEE Intelligent Systems (2007)
- ISWC: IEEE International Symposium on Wearable Computers (2004)
- Pervasive: International Conference on Pervasive Computing (2006-2007)
- Pervasive Computing: IEEE Pervasive Computing (2006)
- SIGGRAPH: ACM Conference on Computer Graphics and Interactive Techniques (2007)
- UbiComp: International Conference on Ubiquitous Computing (2005-2007)
- UIST: ACM Symposium on User Interface Software and Technology (2003-2007)
- UMUAI: User Modeling and User-Adapted Interaction (2006)

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - medium.

Gribble, Steven

Associate professor (2000 Ass't, 2005 Assoc)

Education:

- Ph.D., Computer Science, University of California at Berkeley, 2000
- M.S., Computer Science, University of California at Berkeley, 1997
- B.S., Combined Computer Science and Physics, University of British Columbia, 1995

Other Experience:

- co-founder of ProxiNet, inc. (1997-1998), co-founder of illumita, inc. (2006-2007), technical advisory board member for Isilon Systems.

Consulting, patents, etc.:

- Expert witness for StreamCast, inc. in the RIAA vs. Grokster case, expert witness for the Federal Trade Commission in their first anti-spyware case, expert witness for the office of the attorney general of the state of Washington in their first anti-spyware case.

Principal publications of last five years:

- SpyProxy: On-the-fly Protection from Malicious Web Content, by Alexander Moshchuk, Tanya Bragin, Damien Deville, Steven D. Gribble, and Henry M. Levy. Proceedings of the 16th USENIX Security Symposium (USENIX Security 2007), Boston, MA, August 2007.
- HomeViews: Peer-to-Peer Middleware for Personal Data Sharing Applications, by Roxana Geambasu, Magdalena Balazinska, Steven D. Gribble, and Henry M. Levy. Proceedings of the 2007 SIGMOD International Conference on Management of Data (SIGMOD 2007), Beijing, China, June 2007.
- Why We Search: Visualizing and Predicting User Behavior, by Eytan Adar, Daniel Weld, Brian Bershad, and Steven Gribble. Proceedings of the 16th International World Wide Web Conference (WWW 2007), Banff, Alberta, Canada, May 2007.
- The Importance of History in a Media Delivery System, by Richard Dunn, Henry Levy, Steven Gribble, and John Zahorjan. Proceedings of the 6th International Workshop on Peer-to-Peer Systems (IPTPS 2007), Bellevue, WA, February 2007.
- Cutting through the Confusion: A Measurement Study of Homograph Attacks, by Tobias Holgers, David E. Watson, and Steven D. Gribble. Proceedings of the 2006 USENIX Annual Technical Conference (USENIX '06), Boston, MA, May 2006.
- A Safety-Oriented Platform for Web Applications, by Richard S. Cox, Jacob Gorm Hansen, Steven D. Gribble, and Henry M. Levy. Proceedings of the 2006 IEEE Symposium on Security and Privacy, Oakland, CA, May 2006.
- Presence-Based Availability and P2P Systems, by Richard J. Dunn, John Zahorjan, Steven D. Gribble, and Henry M. Levy. Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing, Konstanz, Germany, August/September 2005.
- Rethinking the Design of Virtual Machine Monitors, by Andrew Whitaker, Richard S. Cox, Marianne Shaw, and Steven D. Gribble. IEEE Computer, Volume 38, number 5, pages 57-62, May 2005.
- Improving the Reliability of Internet Paths with One-hop Source Routing, by Krishna P. Gummadi, Harsha V. Madhyastha, Steven D. Gribble, Henry M. Levy, and David Wetherall. Proceedings of the Sixth Symposium on Operating Systems Design and Implementation (OSDI '04), San Francisco, CA, December 2004.
- Measurement and Analysis of Spyware in a University Environment, by Stefan Saroiu, Steven D. Gribble, and Henry M. Levy. Proceedings of the First Symposium on Networked Systems Design and Implementation (NSDI '04), San Francisco, California, March 2004.

- Dealing with Cheaters in Anonymous Peer-to-Peer Networks, by Paul Gauthier, Brian Bershad, and Steven D. Gribble. University of Washington Technical Report 04-01-03, January 15, 2004.
- Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload, by Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19), Bolton Landing, New York, October 2003.
- The Impact of DHT Routing Geometry on Resilience and Proximity, by Krishna P. Gummadi, Ramakrishna Gummadi, Steven D. Gribble, Sylvia Ratnasamy, Scott Shenker, and Ion Stoica. Proceedings of ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003.
- Mangrove: Enticing Ordinary People onto the Semantic Web via Instant Gratification, by Luke McDowell, Oren Etzioni, Alon Halevy, Henry Levy, Steven D. Gribble, William Pentney, Deepak Verma, and Stani Vlasheva. Proceedings of the Second International Semantic Web Conference (ISWC '03), Sanibel Island, Florida, USA, October 2003.
- Denali: A Scalable Isolation Kernel, by Andrew Whitaker, Marianne Shaw, and Steven D. Gribble. Proceedings of the Tenth ACM SIGOPS European Workshop, Saint-Emilion, France, September 2002.
- Ninja: A Framework for Network Services, by Eric Brewer, Nikita Borisov, Mike Chen, Rob von Behren, Matt Welsh, David Culler, Josh MacDonald, Jeremy Lau, and Steven D. Gribble. Proceedings of the 2002 Usenix Technical Conference, June 2002, Monterey, CA, USA.

Scientific and professional societies of which a member:

- ACM, IEEE, USENIX

Honors and awards:

- ACM Teaching Award (May 2005)
- ACM Teaching Award (May 2001)
- NSF CAREER Award (2002-2007)
- Alfred P. Sloan Research Fellowship (2004-2006)
- Torode Family Endowed Career Development Professorship in Computer Science (2006-2008)

Institutional and professional service in last five years:

- Program Committee Member, ACM SIGCOMM 2007.
- Program Committee Member, 12th International Conference on Architecture Support for Programming Languages and Operating Systems (ASPLOS XII), October 2006.
- Program Committee Member, 2006 IEEE Symposium on Security and Privacy, May 2006.
- Program Committee Member, Third Symposium on Networked Systems Design and Implementation (NSDI '06), May 2006.
- Program Committee Member, 13th Annual Network and Distributed System Security Symposium (NDSS 2006), February 2006.
- Member of the NSF CISE advisory committee, 2003-2005.
- Program Committee Member, Internet Measurement Conference, August 2005.

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Grossman, Dan

Assistant professor (2003 Ass't)

Education:

- Ph.D., Computer Science, Cornell University, 2003
- M.S., Computer Science, Cornell University, 2001
- B.S., Electrical Engineering, Rice University, 1997
- B.A., Computer Science, Rice University, 1997

Principal publications of last five years:

- Enforcing Isolation and Ordering in STM. Tatiana Shpeisman, Vijay Menon, Ali-Reza Adl-Tabatabai, Steve Balensiefer, Dan Grossman, Richard Hudson, Katherine F. Moore, Bratin Saha. ACM Conference on Programming Language Design and Implementation, San Diego, CA, June 2007.
- Searching for Type-Error Messages. Benjamin Lerner, Matthew Flower, Dan Grossman, Craig Chambers. ACM Conference on Programming Language Design and Implementation, San Diego, CA, June 2007.
- Automatically Inferring Structural Changes for Matching Across Program Versions. Miryung Kim, David Notkin, Dan Grossman. 29th International Conference on Software Engineering, Minneapolis, MN, May 2007.
- Safe Manual Memory Management in Cyclone. Nikhil Swamy, Michael Hicks, Greg Morrisett, Dan Grossman, Trevor Jim. Science of Computer Programming, 62(2), pages 122–144, October 2006.
- Quantified Types in an Imperative Language. Dan Grossman. ACM Transactions on Programming Languages and Systems, 28(3), pages 429–475, May 2006.
- Compiling for Template-Based Run-Time Code Generation. Frederick Smith, Dan Grossman, Greg Morrisett, Luke Hornof, Trevor Jim. Journal of Functional Programming, 13(3), pages 677–708, May 2003.
- Preventing Format-String Attacks via Automatic and Efficient Dynamic Checking. Michael F. Ringenburt, Dan Grossman. 12th ACM Conference on Computer and Communications Security, pages 354–363, Alexandria, VA, November 2005.
- AtomCaml: First-Class Atomicity via Rollback. Michael F. Ringenburt, Dan Grossman. 10th ACM International Conference on Functional Programming, pages 92–104, Tallinn, Estonia, September 2005.
- Region-Based Memory Management in Cyclone. Dan Grossman, Greg Morrisett, Trevor Jim, Michael Hicks, Yanling Wang, James Cheney. ACM Conference on Programming Language Design and Implementation, pages 282–293, Berlin, Germany, June, 2002.
- Cyclone: A safe dialect of C. Trevor Jim, Greg Morrisett, Dan Grossman, Michael Hicks, James Cheney, Yanling Wang. USENIX Annual Technical Conference, pages 275–288, Monterey, CA, June 2002.
- Atomicity via Source-to-Source Translation. Benjamin Hindman, Dan Grossman. ACM SIGPLAN Workshop on Memory Systems Performance and Correctness, San Jose, CA, October 2006.
- What Do High-Level Memory Models Mean for Transactions? Dan Grossman, Jeremy Manson, William Pugh. ACM SIGPLAN Workshop on Memory Systems Performance and Correctness, San Jose, CA, October 2006.
- Seminal: Searching for ML Type-Error Messages. Benjamin Lerner, Dan Grossman, Craig Chambers. ACM SIGPLAN Workshop on ML, pages 63–73, Portland, OR, September 2006.
- Types for Describing Coordinated Data Structures. Michael F. Ringenburt, Dan Grossman. ACM SIGPLAN Workshop on Types in Language Design and Implementation, pages 25–36, Long Beach, CA, January 2005.
- Experience With Safe Manual Memory-Management in Cyclone. Michael Hicks, Greg Morrisett, Dan Grossman, Trevor Jim. International Symposium on Memory Management, pages 73–84, Vancouver, Canada, October 2004.
- Type-Safe Multithreading in Cyclone. Dan Grossman. ACM SIGPLAN Workshop on Types in Language Design and Implementation, pages 13–25, New Orleans, LA, January 2003.
- Cyclone: a Type-safe Dialect of C. Dan Grossman, Michael Hicks, Greg Morrisett, Trevor Jim. C/C++

Scientific and professional societies of which a member:

- ACM SIGPLAN

Honors and awards:

- 2nd Place, UW ACM undergraduate teaching award, 2006.
- 1st Place, UW ACM undergraduate teaching award, 2005.
- National Science Foundation CAREER Award, 2005.
- Intel Graduate Student Fellowship, 2002--2003.
- National Science Foundation Fellowship, 1998--2001.
- Best-Paper Award, Colloquium on Principles, Logics, and Implementations of High-Level Programming Languages, 1999.

Institutional and professional service in last five years:

- (Co-)Chair:
- ACM Workshop on Program Analysis for Software Tools and Engineering, 2007.
- Oregon Programming Languages Summer School, 2007.
- Oregon Programming Languages Summer School, 2006
- Program Committee:
- ACM Conference on Object-Oriented Programming Systems, Languages, & Applications 2007
- International Conference on Compiler Construction 2007
- ACM Symposium on Principles of Programming Languages 2007
- ACM Workshop on Partial Evaluation and Program Manipulation 2007
- National Science Foundation Panelist
- Department Committees and service:
- Executive Committee 05-06
- Faculty Recruiting Committee 05-06
- Graduate Admissions Committee 04-05
- Undergraduate Thesis Award Committee 04-05, 05-06
- CSE341 Course Coordinator 05-present

States in which registered: none

Level of activity: professional societies - high; research - high; consulting/summer work in industry - none.

Guruswami, Venkatesan

Assistant professor (2002 Ass't, 2007 Assoc)

Education:

- B. Tech, Computer Science, Indian Institute of Technology, Madras, 1997
- M.S., Computer Science, Massachusetts Institute of Technology, 1999
- Ph.D., Computer Science, Massachusetts Institute of Technology, 2001

Other Experience:

- Miller Research Fellow, UC Berkeley, Sep 2001-August 2002.

Principal publications of last five years:

- V. Guruswami. List decoding of error-correcting codes. Springer, Lecture Notes in Computer Science 3282, 2004.
- V. Guruswami. Algorithmic results in list decoding. NOW publishers, January 2007.
- Surveys:
- V. Guruswami. Iterative Decoding of Low-Density Parity Check Codes. Bulletin of the European Association for Theoretical Computer Science, 90, October 2006.
- V. Guruswami and V. Kabanets. Hardness amplification via space-efficient direct products. Computational Complexity, 2006.
- I. Giotis and V. Guruswami. Correlation clustering with a fixed number of clusters. Theory of Computing, 2(13):249--266, 2006.
- V. Guruswami and A. Rudra. Limits to list decoding Reed-Solomon codes. IEEE Transactions on Information Theory, 52(8):3642--3649, August 2006.
- V. Guruswami and P. Indyk. Linear-time encodable/decodable codes with near-optimal rate. IEEE Transactions on Information Theory, 51(10):3393--3400, 2005.
- V. Guruswami, D. Micciancio, and O. Regev. The complexity of the covering radius problem. Computational Complexity, 14(2):90--121, 2005.
- L. Engebretsen and V. Guruswami. Is constraint satisfaction over two variables always easy? Random Structures and Algorithms, 25(2):150--178, 2004.
- V. Guruswami. Constructions of codes from number fields. IEEE Transactions on Information Theory, 49(3):594--603, 2003.
- V. Guruswami. List decoding from erasures: Bounds and code constructions. IEEE Transactions on Information Theory, 49(11):2826--2833, 2003.
- V. Guruswami, C. Umans, and S. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. IEEE Conference on Computational Complexity, 2007.
- V. Guruswami and P. Raghavendra. A 3-query PCP over integers. ACM Symposium on Theory of Computing, 2007.
- J. Chuzhoy, V. Guruswami, S. Khanna, and K. Talwar. Hardness of routing with congestion in directed graphs. ACM Symposium on Theory of Computing, 2007.
- V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 543--552, October 2006.
- V. Guruswami and A. Rudra. Explicit capacity-achieving list-decodable codes. Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)}, pages 1--10, May 2006.
- V. Guruswami and P. Indyk. Linear-time encodable and list decodable codes. Proceedings of the 35th Annual ACM Symposium on Theory of Computing, pages 126--135, 2003.
- V. Guruswami and P. Indyk. Embeddings and non-approximability of geometric problems. Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 537--538, 2003.
- V. Guruswami and P. Indyk. Near-optimal linear-time codes for unique decoding and new list-decodable

codes over smaller alphabets. Proceedings of the 34th Annual ACM Symposium on Theory of Computing, pages 812--821, 2002.

Scientific and professional societies of which a member:

- Association for Computing Machinery (SIGACT special interest group)

Honors and awards:

- David and Lucile Packard Fellowship for Science and Engineering, 2005. (One out of 16 fellows.)
- Alfred P. Sloan Foundation Fellow, 2005.
- NSF Faculty Early Career Development (CAREER) Award, 2004.
- Association for Computing Machinery (ACM) Doctoral Dissertation Award, 2002.
- George M. Sprowls Award, MIT, 2002.
- Miller Research Fellow, 2001.
- IEEE Information Theory Society Paper Award, 2000.
- IBM Research Fellowship for the academic year 1999-2000.
- AT&T Leadership Award 1997.
- 2nd position in the All India Joint Entrance Examination, 1993.
- 3rd position, Indian National Mathematical Olympiad, 1992.

Institutional and professional service in last five years:

- Organizer, Minisymposium on coding theory, SIAM Conference on Discrete Mathematics, June 2006.
- Guest Editor (together with Valentine Kabanets), Computational Complexity. Special issue on selected papers from CCC 2006 -- the 21st IEEE Conference on Computational Complexity.
- Guest Editor (together with Edith Cohen), Journal of Computer and System Sciences, 68(4), June 2004. Special issue on selected papers from the 43rd Annual IEEE Symposium on Foundations of Computer Science, November 2002.
- IEEE Symposium on Foundations of Computer Science, 2005.
- IEEE International Symposium on Information Theory, 2006.
- Foundations of Software Technology and Theoretical Computer Science, 2005.
- IEEE Symposium on Foundations of Computer Science, 2002.
- Refereed journal papers for Combinatorica, IEEE Transactions on Information Theory, SIAM Journal on Computing, Information Processing Letters, SIAM J. on Discrete Mathematics, Discrete Mathematics, Theory of Computing, and Journal of the ACM.

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Hemingway, Bruce

Lecturer (2002 Ass't)

Education:

- B.A., Music, Indiana University, 1973

Other Experience:

- Academic Experience:
- 2002-present:
- Lecturer, Department of Computer Science & Engineering, University of Washington
- Courses taught:
- CSE 466, Software for Embedded Systems
- CSE 370, Introduction to Digital Design
- CSE P567, Design & Implementation of Digital Systems
- CSE 467, Advanced Digital Design
- 1973-1982: Adjunct Lecturer, Division of Music, Indiana University at South Bend
<http://www.iusb.edu/~sbarts/>
- Courses taught For Division of Music:
- A490 Large-Scale Electronic Music Composition
- R372 Multi-media Production
- G324 Synthesizer Performance Techniques
- A300 Introduction to Electronic Studio Techniques
- M201 Literature & Styles of Music: 17th and 18th centuries
- M202 Literature & Styles of Music: 19th and 20th centuries
- M174 Music Appreciation
- Courses taught for Division of Continuing Education:
- Personal Computers for Non-Professionals
- Machine-Language Programming for Microcomputers
- 1968-1973: Technical Director and Administrative Assistant to the Chairman, Division of Music, Indiana University at South Bend
- Full-time Staff position; duties included:
- Design and technical direction of the Electronic Music Studio
- Technical Direction of all musical performances (more than 300), including 80 touring chamber opera performances, and productions of Lucia and The Barber of Seville.
- Administrative responsibilities included student counseling and performance scheduling.
- Education: A.B. in Music (emphasis in Electronic Music Composition and Orchestral Conducting) Indiana University, 1973

Consulting, patents, etc.:

- Technical Experience:
- 2002-present: Manager, Computer Engineering Laboratory, Department of Computer Science & Engineering, University of Washington
- 1993-2002: Co-founder, CEO, and digital designer, dB Technologies, Inc., Seattle, WA. Manufacturer of high-end professional analog-to-digital and digital-to-analog audio converters for the recording and film industries.
- 1986-1993: Independent Consultant in digital audio hardware and software design.
- Some major projects:
- Digital Signal Processing software for the Pro Spatializer from Spatializer Audio Laboratories, Inc., Woodland Hills, CA. This device allows the spatial placement, movement, and scaling of individual tracks

in space from two loudspeakers. The device works in real-time and processes up to 24 simultaneous sounds, each individually placed in a stereo sound field. <http://spatializer.com/>

- Designed prototype of DPR-100 Digital Audio Workstation, a Disk-based Recorder-Editor, for Symetrix, Inc., Lynnwood, Washington. These devices (there were several envisioned models in the product line) were high-end products intended for recording studios, film studios, and video post-production. <http://www.symetrixaudio.com/>
- In 1988, for start-up Auris, Inc., Evanston, Ill., a prototype multiprocessor Spatial Processor, based on and implementing the patented algorithms of Gary Kendall of Northwestern University. This was a Digital Signal Processor- based design that allowed the real-time three-dimensional manipulation of sounds from stereo speakers using 34 Motorola 56000-series digital signal processors. Two prototype systems were built for Auris. <http://www.nwu.edu/musicschool/personal/kendall/kendall.html>
- 1982-1986: Director of Research and Development for Digital Systems International, Redmond, Washington (later Mosaix) Designed Voice-recognition-based call-management systems for telemarketing and credit-collection management.
- 1977-1982: Design Engineer- Consulting Electronic Engineering Firm: Environmental Technology, Inc., South Bend, Indiana <http://www.networketi.com/> Numerous embedded system design projects
- 1973-1977: Vice President and General Manager- I.E. Inc., South Bend, Indiana, Multimedia Production Company producing industrial educational training films and videos

Principal publications of last five years:

- Papers:
- Bruce Hemingway, Waylon Brunette, Gaetano Borriello, "Variations on the Flock: Wireless Embedded Systems in an Undergraduate Curriculum", Proceedings of the Australian Telecommunications, Networks and Applications Conference (ATNAC) 2006, Melbourne, Australia, December, 2006, pp. 366-370 (Invited Keynote Paper)
- .
- Bruce Hemingway, Waylon Brunette, Tom Anderl, Gaetano Borriello: The Flock: Mote Sensors Sing in Undergraduate Curriculum. IEEE Computer 37(8): 72-78 (2004)

States in which registered: none

Level of activity: professional societies - low; research - low; consulting/summer work in industry - none.

Karlin, Anna

Professor (1996 Ass't, 1996 Assoc, 1998 Full)

Education:

- Sc.B., Mathematical Sciences, Stanford University, 1981
- Ph.D., Computer Science, Stanford University, 1987

Other Experience:

- 1988 -- 1994 Principal Scientist, Digital Equipment Corporation's
- Systems Research Center. Palo Alto, CA.

Consulting, patents, etc.:

- Patent No. 5032987, "Dynamic Hashing with Multiple Tables" (with A. Broder), 1990.
- Patent Application Pending, OTT Ref # 2035-2527, "Using Global Memory Information to Manage Memory in a Computer Network" (with M. Feeley, W. Morgan, F. Pighin, H. Levy, and C. Thekkath), 1997.

Principal publications of last five years:

- Competitive Auctions, with A. Goldberg, J. Hartline, M. Saks and A. Wright, Games and Economic Behavior, Special Issue on Electronic Market Design, Volume 55, Issue 2, 2006.
- Generalized Competitive Auctions, with A. Fiat, A. Goldberg and J. Hartline, In Proceedings of 34th ACM Symposium on Theory of Computing, 2002.
- Truthful and Competitive Market Clearing, with K. Deshmukh, A. Goldberg and J. Hartline, Proceedings of 2002 European Symposium on Algorithms.
- A Lower Bound on the Competitive Ratio of Truthful Auctions, with A. Goldberg, J. Hartline and M. Saks, In 21st Symposium on Theoretical Aspects of Computer Science, 2004.
- On Profit-Maximizing Envy-Free Pricing, with V. Guruswami, J. Hartline, D. Kempe, C. Kenyon and F. McSherry, In Proceedings of 16th ACM Symposium on Discrete Algorithms, 2005.
- Beyond VCG: Frugality in Truthful Mechanisms, with D. Kempe and T. Tamir, In Proceedings of 46th IEEE Conference on Foundations of Computer Science, 2005.
- Cheap Labor Can be Expensive, with N. Chen, In Proceedings of 18th ACM Symposium on Discrete Algorithms, 2007.
- Greedy Bidding Strategies for Keyword Auctions, with M. Cary, A. Das, B. Edelman, I. Giotis, K. Heimerl, C. Mathieu and M. Schwarz. In ACM Conference on Electronic Commerce, 2007.

Scientific and professional societies of which a member:

- ACM
- SIGACT

Honors and awards:

- Outstanding Paper Award, Proceedings of ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, 1995.
- Invited Speaker:
- Simon Fraser University Distinguished Lecturer 2005, MIT CSAIL Distinguished Lecturer 2004, Online Algorithms 2004, Workshop on "Auction Theory and Practice" 2003, Duke/UNC/NCState Computer

Science Distinguished Lecturer 2003, Johns Hopkins Department of Computer Science Distinguished Lecturer 2003, AIRO 2002, Rome Theory Day 2002, ALGO 2002 (ESA/APPROX/WABI), ALLENEX 2001, RANDOM/APPROX 2001, UAI 2001, PACT Theory Day 2001, 1997 Grace Hopper Celebration of Women in Computing, Jerusalem Combinatorics '93, Institute for Mathematics and Applications workshop on "Finite Markov Chain Renaissance", 1993.

Institutional and professional service in last five years:

- Member of Editorial Board, SIAM Journal on Computing
- Member, NRC Computer Science and Telecommunications Board, 1999-2002
- Member, ACM Paris Kanellakis Theory and Practice Award Subcommittee 2004- 2007
- Program Committee member:
- IEEE Foundations of Computer Science 2003,
- ACM Conference on Electronic Commerce 2006,
- ACM World Wide Web Conference (Search Track) 2006,
- IEEE Foundations of Computing 2006,
- ACM Conference on Electronic Commerce 2007,
- Third Workshop on Sponsored Search 2007.
- Graduate Program Coordinator, Department of Computer Science.
- Served as referee for Journal of the ACM, Algorithmica, Theoretical
- Computer Science, SIAM Journal on Computing, Combinatorica,
- Transactions on Computer Systems, Random Structures and Algorithms, etc.

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Kohno, Tadayoshi

Assistant professor (2006 Ass't)

Education:

- Ph.D., Computer Science, University of California at San Diego, 2006
- M.S., Computer Science, University of California at San Diego, 2004
- B.S., Computer Science, University of Colorado, 1999

Other Experience:

- Cryptographer, Software Security Group, Cigital, January 2000--June 2001.
- Cryptographer, Counterpane Systems, May 1999--January 2000.

Principal publications of last five years:

- JOURNAL PUBLICATIONS (LAST 5 YEARS)
- John Viega, J. T. Bloch, Tadayoshi Kohno, and Gary McGraw. Token-based scanning for source code security problems. *ACM Transactions on Information and System Security*, 5(3):238--261, August 2002.
- Mihir Bellare, Tadayoshi Kohno, and Chanathip Namprempre. Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the Encode-then-Encrypt-and-MAC paradigm. *ACM Transactions on Information and System Security*, 7(2):206--241, May 2004.
- Tadayoshi Kohno, Andre Broido, and K.C. Claffy. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93--108, April--June 2005.
- Mihir Bellare, Tadayoshi Kohno, and Chanathip Namprempre. Authenticated encryption in SSH: Provably fixing the SSH binary packet protocol. In Vijay Atluri, editor, *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 1--11. ACM Press, November 2002.
- Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In Eli Biham, editor, *Advances in Cryptology -- EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 491--506. Springer-Verlag, May 2003.
- Tadayoshi Kohno, John Viega, and Doug Whiting. CWC: A high-performance conventional authenticated encryption mode. In Bimal Roy and Willi Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 408--426. Springer-Verlag, February 2004.
- Tetsu Iwata and Tadayoshi Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. In Bimal Roy and Willi Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 427--445. Springer-Verlag, February 2004.
- Mihir Bellare and Tadayoshi Kohno. Hash function balance and its impact on birthday attacks. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology -- EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 401--418. Springer-Verlag, May 2004.
- Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach. Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy*, pages 27--40. IEEE Computer Society, May 2004.
- Tadayoshi Kohno. Attacking and repairing the WinZip encryption scheme. In Birgit Pfitzmann, editor, *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 72--81. ACM Press, October 2004.
- Tadayoshi Kohno, Andre Broido, and kc claffy. Remote physical device fingerprinting. In *IEEE*
- Naveen Sastry, Tadayoshi Kohno, and David Wagner. Designing voting machines for verification. In *15th Usenix Security Symposium*, August 2006.
- Mihir Bellare, Tadayoshi Kohno, and Victor Shoup. Stateful public-key cryptosystems: How to encrypt with one 160-bit exponentiation. In Rebecca Wright and Sabrina De Capitani di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security*. ACM Press, November 2006.

- Ben Greenstein, Ramakrishna Gummadi, Jeffrey Pang, Mike Y. Chen, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. Can Ferris Bueller still have his day off? Protecting privacy in the wireless era. In 11th Workshop on Hot Topics in Operating Systems, May 2007.
- T. Scott Saponas, Jonathan Lester, Carl Hartung, Sameer Agarwal, and Tadayoshi Kohno. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In 16th Usenix Security Symposium, August 2007.

Scientific and professional societies of which a member:

- International Association for Cryptologic Research (IACR), 1999--Present
- Association for Computing Machinery (ACM), 2006--Present
- Institute of Electrical and Electronics Engineers (IEEE), 2006--Present

Honors and awards:

- IBM Ph.D. Fellowship, 2004.
- National Defense Science and Engineering Graduate Fellowship, 2001.
- Cal IT Fellowship, University of California at San Diego, 2001. Declined in favor of the above fellowship.
- CSE PhD Dissertation Award, 2006, University of California at San Diego, Department of Computer Science and Engineering.
- Award paper, 2005 IEEE Symposium on Security and Privacy; paper invited to a special issue of the IEEE Transactions on Dependable and Secure Computing.

Institutional and professional service in last five years:

- Usenix Security, San Jose, CA, August 2008.
- IEEE Symposium on Security and Privacy, Oakland, CA, May 2008
- Advances in Cryptology -- EUROCRYPT, Istanbul, Turkey, April 2008.
- IEEE Workshop on Mobile Computing Systems and Applications (HotMobile), Sonoma or Napa, CA, February, 2008.
- WWW 2007 Security, Privacy, Reliability and Ethics Track, Banff, Canada, May 2007.
- ACM Workshop on Recurring Malcode (WORM), Fairfax, VA, November 2006.
- ACM Workshop on Wireless Security (WiSe 2006), Los Angeles, CA, September 2006.
- Financial Cryptography and Data Security, Anguilla, BWI, February 2006.
- Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2006.
- Usenix Security 2005, Baltimore, MD, August 2005.
- WWW 2005 Security, Privacy, and E-Commerce Track, Chiba, Japan, May 2005.
- WWW 2004 Security and Privacy Track, New York, NY, May 2004.

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Krishnamurthy, Arvind

Assistant professor (2005 Ass't)

Education:

- Bachelor of Technology, Computer Science, IIT, Madras, 1991
- M.S., Computer Science, UC Berkeley, 1994
- Ph.D., Computer Science, UC Berkeley, 1999

Other Experience:

- Assistant Professor, Yale University, 1999-2005
- Principal Scientist, FirstRain Inc, 2000-2002

Principal publications of last five years:

- Do incentives build robustness in BitTorrent? M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, A. Venkataramani To appear in NSDI, 2007.
- Leveraging BitTorrent for End Host Measurements. T. Isdal, M. Piatek, A. Krishnamurthy, T. Anderson. To appear in PAM, 2007.
- Towards IP Geolocation using Delay and Topology Measurements. E. Katz-Bassett, J. John, A. Krishnamurthy, D. Wetherall, T. Anderson, Y. Chawathe. Appeared in IMC, 2006.
- A Structural Approach to Latency Prediction. H. Madhyastha, T. Anderson, A. Krishnamurthy, N. Spring, A. Venkataramani. Appeared in IMC, 2006.
- iPlane: An Information Plane for Distributed Services. H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, A. Venkataramani. Appeared in OSDI, 2006.
- Optimal Capacity Sharing of Network with Multiple Overlays. Z. Ma, J. Chen, R. Yang, A. Krishnamurthy. Appeared in IWQoS, 2006.
- PCP: Efficient Endpoint Congestion Control. T. Anderson, A. Collins, A. Krishnamurthy, J. Zahorjan. NSDI, 2006.
- Secure Routing. I. Avramopoulos, H. Kobayashi, A. Krishnamurthy, R. Wang. To appear in Network Security, Wiley.
- Oasis: An Overlay-Aware Network Stack. H. Madhyastha, A. Venkataramani, A. Krishnamurthy, T. Anderson. Operating Systems Review, 2006.
- Combining Flexibility and Scalability in a Peer-to-Peer Publish/Subscribe System. C. Zhang, A. Krishnamurthy, and R. Wang. Appeared in Middleware, 2005.
- Stable Route Selection for Interdomain Traffic Engineering. R. Yang, H. Xie, H. Wang, L. Li, Y. Liu, A. Silberschatz, and A. Krishnamurthy. Appeared in IEEE Network, 2005.
- Network Localization in Partially Localizable Networks. D. Goldenberg, A. Krishnamurthy, W. Maness, R. Yang, A. Young, and A. Savvides. Appeared in Infocom 2005.
- A Collision Model for Randomized Routing in Fat-Tree Networks. V. Strumpfen and A. Krishnamurthy. Appeared in Journal of Parallel and Distributed Computing.
- Brushwood: Distributed Trees in Peer-to-Peer Systems. C. Zhang, A. Krishnamurthy, and R. Wang. Appeared in IPTPS, 2005.
- Bridging the digital divide: storage media + postal network = generic high-b/w communication. N. Garg, S. Sobti, F. Zheng, J. Lai, K. Li, A. Krishnamurthy, and R. Wang. Appeared in ACM Transactions on Storage Systems, 2005.
- Turning the Postal System into a Generic Digital Communication Mechanism. R. Wang, N. Garg, S. Sobti, E. Ziskind, J. Lai, and A. Krishnamurthy. Appeared in Sigcomm 2004.
- Managing a Portfolio of Overlay Paths. D. Antonova, A. Krishnamurthy, Z. Ma, and R. Sundaram. Appeared in NOSSDAV, 2004.
- Load Balancing and Locality in Range-Queriable Data Structures. J. Aspnes, J. Kirsch, and A.

Krishnamurthy. Appeared in PODC, 2004.

- A Transport Layer Approach for Improving End-to-End Performance Using Redundant Paths. M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang. Appeared in Usenix Annual Technical Conference, 2004.
- Distance Learning Technologies for Basic Education in Disadvantaged Areas. Appeared in GCCCE 2004.
- Segank: A Distributed Mobile Storage System. S. Sobti, N. Garg, F. Zheng, J. Lai, A. Krishnamurthy, and R. Wang. Appeared in Usenix Conference on File and Storage Technologies, 2004.
- Network-Embedded Programmable Storage and Its Applications. S. Sobti, J. Lai, Y. Shao, N. Garg, C. Zhang, F. Zheng, A. Krishnamurthy, and R. Wang. Appeared in IFIP Conference on Networking, 2004.
- Overlay Mesh Construction Using Interleaved Spanning Trees. A. Young, J. Chen, Z. Ma, A. Krishnamurthy, L. Peterson, and R. Wang. Appeared in Infocom 2004.
- Highly Secure and Efficient Routing. I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy. Appeared in Infocom 2004.

Honors and awards:

- National Science Foundation Career Grant: "Compiler Aided Resource Management for Clusters." Principal Investigator. Duration: March 2000-2005.
- National Science Foundation Grant: "Incentive-Compatible Designs for Distributed Systems." co-Principal Investigator. Duration: June 2002-2006.
- National Science Foundation Grant: "Composing Data-Rich Embedded Systems the Easy Way." Principal Investigator. Duration: July 2002-2004.
- National Science Foundation Grant: "MRI Development: Enabling Lightweight Planetary-Scale Services." co-Principal Investigator. Duration: July 2006-2010.
- National Science Foundation Grant: "A Prototype GENI Authorization Service." co-Principal Investigator. Duration August 2006-2007.

Institutional and professional service in last five years:

- Program-Committee Member, Internet Measurement Conference, 2007.
- Program-Committee Member, GlobeCom, 2007.
- Program-Committee Member, IWQoS, 2006.
- NSF panel review member for Distributed Systems and Operating Systems Program, 2005.
- NSF panel review member for Distributed Systems and Operating Systems Program, 2004.
- Program-Committee Member, {it International Conference on Distributed Computing Systems}, 2003.
- Program-Committee Member, {it International Conference on Parallel Processing}, 2003.
- NSF panel review member for the Computer Systems Architecture Program, 2003.
- Program-Committee Member, High-Performance Computing for Computational Biology, 2002.

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - none.

Ladner, Richard

Professor (1971 Ass't, 1976 Assoc, 1981 Full)

Education:

- Ph.D., Mathematics, University of California, Berkeley, 1971
- B.S., Mathematics, St. Mary's College of California, 1965

Other Experience:

- Visiting Positions:
- University of Toronto, 1977
- Yale University, 1978
- Gallaudet University, 1985
- Mathematical Sciences Institute, Berkeley, 1986
- Victoria University, Wellington, New Zealand, 1993
- AT&T Labs, Florham Park, NJ, 1999-2000

Consulting, patents, etc.:

- Microsoft
- IBM
- AT&T

Principal publications of last five years:

- A. Cavender, D.K. Barney, R.E. Ladner, R. Vanam, E.A. Riskin. MobileASL: Intelligibility of Sign Language Video as Constrained by Mobile Phone Technology. To appear in Disability and Rehabilitation: Assistive Technologies, ASSETS 2006 Special Issue.
- J. Goshi and R. E. Ladner. Algorithms for Dynamic Multicast Key Distribution Trees. ACM Journal on Experimental Algorithmics, Vol 11, Article 1.4, 2006.
- S. Pan, G. Shavit, M. Penas-Centeno, D.-H. Xu, L. Shapiro, R. Ladner, E. Riskin, W. Hol, D Meldrum. Automated Classification of Protein Crystallization Images Using Support Vector Machines with Scale-Invariant Texture and Gabor Features. {em Acta Crystallographica Section D}, 2006, 271-279.
- A. Bar-Noy, J. Goshi, R.E. Ladner. Off-line and On-line Guaranteed Start-up Delay for Media-on-Demand with Stream Merging. Journal of Discrete Algorithms, vol. 4, No. 1, 2006, 72-105 .
- A. Bar-Noy and R.E. Ladner. Efficient Algorithms for Optimal Stream Merging for Media-on-Demand. SIAM Journal on Computing, Vol. 33, No. 5, 2004, 1011-1034.
- R. Anderson, S. Kannan, H. Karloff, and R.E. Ladner. Thresholds and Optimal Binary Comparison Search Trees. Journal of Algorithms, Vol 44, 2002, 338-358.
- Book Chapters:
- J. Bigham and R.E. Ladner. Accessmonkey: A Collaborative Scripting Framework for Web Users and Developers. To appear International Cross-Disciplinary Conference on Web Accessibility, 2007 (W4A, 2007).
- S. Kane, T.J. Shockley, J. Shulman, R.E. Ladner. A Web Accessibility Report Card for Top International University Web Sites. To appear in International Cross-Disciplinary Conference on Web Accessibility, 2007 (W4A, 2007).
- R. Vanam, E.A. Riskin, S.S. Hemami, R.E. Ladner. Distortion-Complexity Optimization of the H.264/MPEG-4 AVC Encoder using the GBFOS Algorithm. To appear Data Compression Conference 2007.
- A. Bar-Noy, R.E. Ladner, T. Tamir. Optimal Delay for Media-on-Demand with Pre-loading and Pre-buffering. 13th Colloquium on Structural Inforamtion and Communication 13th Colloquium on Structural

Information and Communication Complexity (SIROCCO 2006), 171-181.

- A. Bar-Noy, J. Christensen, R.E. Ladner, T. Tamir. A General Buffer Scheme for the Windows Scheduling Problem. WEA 2005: 4th International Workshop on Efficient and Experimental Algorithms, 2005, 139-151.
- Scheduling Problems for Broadcast Systems. Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA 2002), January 2002, 433-442.
- A. Bar-Noy, G. Goshi, R.E. Ladner, and K. Tam. Comparison of Stream Merging Algorithms for Media-on-Demand. Multimedia Computing and Networking (MMCN'02), January 2002.

Scientific and professional societies of which a member:

- ACM and SIGACT
- IEEE Computer Society
- American Mathematical Society
- Association for Symbolic Logic

Honors and awards:

- Fulbright, 1992-93
- Guggenheim, 1985-86
- Fellow of ACM, 1994-
- Boeing Professor in Computer Science and Engineering, 2004 -
- Presidential Award for Excellence in Science, Mathematics and Engineering Mentoring (PAESMEM), 2004
- Japan Society for the Promotion of Science Fellow, 2006

Institutional and professional service in last five years:

- Member of the NSF Committee on Equal Opportunities in Science and Engineering (CEOSE), 2006 -
- Member of the NSF CISE Advisory Committee, 2006 -
- External Review Committee for Computer Sciences at Purdue University, 2005
- Chair of ACM SIGACT, 2005 -
- Knuth Prize Committee Member, ACM SIGACT and IEEE TCMFC, 2003-
- NSF Review Panel 2003
- Fourth International Cross-Disciplinary Conference on Web Accessibility (W4A 2007)
- The Ninth International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2007)
- Member, Search Committee for the Dean of Engineering, 2005-06
- Member, Dean of Arts and Sciences Review Committee, 2005
- Member, EE Chair Search Committee, 2004
- Program Coordinator for Professional Masters Program, 2000-2005
- Organizer of the Vertical Mentoring Workshop for the Blind in Science, Technology, Engineering, and Science, 2006

States in which registered: none

Level of activity: professional societies - high; research - high; consulting/summer work in industry - low.

Landay, James

Associate professor (1997 Ass't, 2002 Assoc)

Education:

- B.S., Electrical Engineering & Computer Science, University of California, Berkeley, 1990
- M.S., Computer Science, Carnegie Mellon University, 1993
- Ph.D., Computer Science, Carnegie Mellon University, 1996

Other Experience:

- Laboratory Director of Intel Research Seattle, 8/2003-8/2006
- UC Berkeley faculty, 1997-2003
- NetRaker Corp, chief scientist & CTO, 1999-2003

Consulting, patents, etc.:

- Consultant with Fuji-Xerox Palo Alto Labs, Intel Corp., Fish & Richardson (legal), Klipp, Colussy, Jenks, DuBois (architects), Pangea Systems, Propel Software, SkyFlow

Principal publications of last five years:

- Yang Li, Jason I. Hong, James A. Landay, "Design Challenges and Principles for Wizard of Oz Testing of Location-Enhanced Applications." In IEEE Pervasive Computing Magazine, 6(2), April-June 2007, pp. 70-75.
- Scott Lederer, Jason I. Hong, Anind Dey, and James A. Landay, "Personal Privacy through Understanding and Action: Five Pitfalls for Designers." In Personal and Ubiquitous Computing, 8(6): pp. 440-454, November 2004.
- Susumu Harada, James A. Landay, Jonathan Malkin, Xiao Li, and Jeff A. Bilmes. "The vocal joystick: evaluation of voice-based cursor control techniques." In Proceedings of ASSETS, pp. 197-204, ACM, October 2006.
- X. Li, J. Malkin, S. Harada, J. Bilmes, R. Wright and J. Landay. "An Online Adaptive Filtering Algorithm for the Vocal Joystick." In Proceedings of Interspeech. Pittsburgh, PA, September 2006.
- J. Bilmes, J. Malkin, X. Li, S. Harada, K. Kilanski, K. Kirchhoff, R. Wright, A. Subramanya, J. Landay, P. Dowden, and H. Chizeck. "The Vocal Joystick." IEEE Intl. Conf. on Audio, Speech and Signal Processing. Toulouse, France, May 2006.
- Yang Li and James A. Landay, "Informal Prototyping of Continuous Graphical Interactions." In Proceedings of UIST 2005: ACM Symposium on User Interface Software and Technology, Seattle WA, 2005.
- Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay, "Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces." In Proceedings of CHI 2005: ACM Conference on Human Factors in Computing Systems, Portland OR, 2005.
- Jeffrey Heer, Stuart K. Card, and James A. Landay, "prefuse: a toolkit for interactive information visualization." In Proceedings of CHI 2005: ACM Conference on Human Factors in Computing Systems, Portland OR, 2005
- Jason I. Hong and James A. Landay, "An Architecture for Privacy-Sensitive Ubiquitous Computing." In Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (Mobisys 2004). Boston, MA, pp. 177-189, 2004 [14% acceptance rate].
- . Hong, Kevin Wang, Leila A. Takayama, and James A. Landay. "Siren: Context-aware Computing for Firefighting." In the Proceedings of Second International Conference on Pervasive Computing (Pervasive 2004), Vienna, Austria, pp. 87-105 [13% acceptance rate].
- , "Books with Voices: Paper Transcripts as a Tangible Interface to Oral Histories." In the Proceedings of

CHI 2003, ACM Conference on Human Factors in Computing Systems, CHI Letters, 5(1), pp. 89-96 [16% acceptance rate].

- Miriam Walker, Leila Takayama, James A. Landay, "High-fidelity or low-fidelity, paper or computer medium?" In the Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting, Baltimore, October 2002, pp. 661-665.
- Scott R. Klemmer, Michael Thomsen, Ethan Phelps-Goodman, and James A. Landay. "Where Do Web Sites Come From? Capturing and Interacting with Design History." In Proceedings of CHI 2002, ACM Conference on Human Factors in Computing Systems, CHI Letters, 4(1), pp. 1-8. [15% acceptance rate].

Scientific and professional societies of which a member:

- ACM
- IEEE

Honors and awards:

- Computer Science Division Information Technology Faculty Award (2003)
- National Science Foundation CAREER Award (1999)

Institutional and professional service in last five years:

- UW CSE department admissions committee (2007)
- UW CSE human subjects board liaison (2006-2007)
- IEEE Pervasive Computing (2004-2007)
- Conference & Program Committees
- CHI: ACM Conference on Human Factors in Computing Systems
- Papers Program Committee (2001-2002, 2004)
- ICMI: International Conference on Multimodal Interfaces Program Committee (2003)
- Intel Research Symposium General Co-Chair (2006)
- IUI: International Conference on Intelligent User interfaces Program Committee (2007)
- UIST: ACM Symposium on User Interface Software & Technology: Program Chair (2004)
- Surveys Chair (2000)
- Program Committee (1999, 1998, 2003, 2007)
- ACM Transactions on Computer-Human Interaction (1998, 2003)
- ACM SIGMOBILE Mobile Computing and Communications Review (2002)
- CHI: Human Factors in Computing Systems (1995-2000, 2003)
- Computers & Graphics (2000)
- Eurographics Workshop on Design, Specification, Verification of Interactive Systems (1995, 1996)
- Eurographics Workshop on Sketch-Based Interfaces and Modeling (2006)
- NSF HCC Program Panel Reviewer (2007)

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - high.

Lazowska, Edward D.

Professor (1977 Ass't, 1982 Assoc, 1986 Full)

Education:

- A.B., Independent Major, Brown University, 1972
- M.Sc., Computer Science, University of Toronto, 1974
- Ph.D., Computer Science, University of Toronto, 1977

Other Experience:

- Visiting Scientist, DEC Systems Research Center, 1984-85
- Visiting Scholar, Computer Science Department, Stanford University, 1984-85
- Visiting Scholar, Department of Computer Science & Engineering, UC San Diego, 2001-02

Consulting, patents, etc.:

- Corporate Technical Advisory Boards: Microsoft Research, 1991-; Voyager Capital, 1996-; Ignition Partners, 2000-; Frazier Technology Ventures, 2000-06; Madrona Venture Group, 2000-; Impinj, 2001-; Conenza, 2007-
- Corporate Boards of Directors: Data I/O Corp., 1996-; Intrepid Learning Solutions, 1999-
- Organizational Boards of Directors: Washington Software Alliance, 1996-; Computing Research Association, 1995-2003 (Chair, 1997-2001; Chair, Government Affairs Committee, 1992-97 and 2001-03); Technology Alliance of Washington, 2000- (Executive Committee, 2001-; Vice President, 2001-03; Council member, 1995-2000); ACM Council, 2000-03; LakesidemSchool, 2001-04; Washington
- Digital Learning Commons, 2003-; Pacific Northwest GigaPoP, 1999-; Pacific Wave, 2001-
- State of Washington Information Services Board, 1995-

Principal publications of last five years:

- B. St. Arnaud, A. Chave, A. Maffei, E Lazowska, L. Smarr, and G. Gopalan. An Integrated Approach to Ocean Observatory Data Acquisition/Management and Infrastructure Control Using Web Services. *Marine Technology Society Journal* (2004).
- E. Lazowska and D. Patterson. An Endless Frontier Postponed. *Science* 308, 6 May 2005.
- E. Lazowska and D. Patterson. Computing Research: A Looming Crisis. *ACM SIGCOMM Computer Communication Review* 35,3, July 2005.
- D. Richardson, S. Gribble, and E. Lazowska. The Limits of Global Scanning Worm Detectors in the Presence of Background Noise. *Proc. 3rd Workshop on Rapid Malcode* (Nov. 2005).
- T. Parikh and E. Lazowska. Designing an Architecture for Delivering Mobile Information Services to the Rural Developing World. *Proc. World Wide Web Conf. 2006* (May 2006).
- E. Lazowska and J. Zahorjan. In Memoriam: Kenneth C. Sevcik. *Proc. ACM SIGMETRICS / Performance '06*, June 2006.

Scientific and professional societies of which a member:

- National Academy of Engineering
- American Academy of Arts & Sciences (Fellow)
- Association for Computing Machinery (Fellow)
- Institute of Electrical and Electronics Engineers (Fellow)
- American Association for the Advancement of Science (Fellow)

- IFIP Working Group 7.3 on Computer System Modeling

Honors and awards:

- Association for Computing Machinery Presidential Award, 2005
- Computing Research Association Distinguished Service Award for outstanding service to the computing research community, 2005
- Named one of "Seattle's 25 most influential people," Seattle magazine, 2004
- Fellow of the American Academy of Arts & Sciences, 2004
- Named to "GT 25" (national leaders of information technology in state government) by Government Technology magazine, 2002
- Elected to the National Academy of Engineering, 2001
- Fellow of the American Association for the Advancement of Science, 2001
- Bill & Melinda Gates Endowed Chair in Computer Science & Engineering, 2000
- University of Washington Brotman Award for Instructional Excellence (Departmental), 1999
- University of Washington Outstanding Public Service Award, 1998
- Alliance for Education "A+ Partnership Award" for Outstanding Contributions to the Seattle Public Schools, 1997
- Award paper, 1993 Machnix Workshop
- Award Paper, 13th ACM Symposium on Operating Systems Principles (1991)
- Award Paper, 12th ACM Symposium on Operating Systems Principles (1989)
- Award paper, 1989 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems

Institutional and professional service in last five years:

- Professional:
- National Research Council Committee on Assessing the Impacts of Changes in the Information Technology Research and Development Infrastructure, 2006-07
- Chair, Computing Research Association GENI Community Advisory Board, 2006-07
- Co-chair, President's Information Technology Advisory Committee, 2003-05
- Chair, Information Science and Technology (ISAT) Study Group of the Defense Advanced Research Projects Agency, 2004-06; Vice Chair (Chair-designate), 2002-04; Member, 1998-2001
- National Academy of Engineering Section 5 (Computer Science & Engineering) Peer Committee, 2002-06; Chair (and member of the NAE Committee on Membership), 2004; Vice Chair (Chair-designate), 2003
- National Academy of Engineering Section 5 (Computer Science & Engineering) Vice Chair (Chair-designate), 2006-07
- American Academy of Arts & Sciences, Membership Panel, Class I Section 6 (Computer Sciences), 2006-
- American Association for the Advancement of Science Section T (Information, Computing, and Communication) Steering Group, 2003-; ACM liaison to AAAS, 2002-08
- Institutional:
- Chair, Electrical Engineering Chair Search Committee, 2007
- Information School Founding Board, 2004-
- Schidler Center for Law, Commerce and Technology Advisory Committee, 2001-
- External Relations and Development Coordinator, Dept. of Computer Science & Engineering, 2001-

States in which registered: none

Level of activity: professional societies - high; research - low; consulting/summer work in industry - high.

Lee, James

Assistant professor (2006 Ass't)

Education:

- Ph.D., Computer Science, University of California, Berkeley, 2005
- B.S., Computer Science, Purdue University, 2001
- B.S., Mathematics, Purdue University, 2001

Principal publications of last five years:

- Euclidean distortion and the Sparsest Cut (STOC 2005 and Journal of the AMS)
- Improved approximation algorithms for minimum-weight vertex separators (STOC 2005 and SIAM J. on Computing)
- Algorithms on negatively curved spaces (FOCS 2006)
- L_p metrics on the Heisenberg group and the Goemans-Linial conjecture (FOCS 2006)
- Trees and Markov convexity (SODA 2006)
- Vertex cuts, random walks, and dimension reduction in series-parallel graphs (STOC 2007)
- Volume distortion for subsets of Euclidean spaces (SoCG 2006)
- Distance scales, embeddings, and metrics of negative type (SODA 2006)
- An improved approximation ratio for the minimum linear arrangement problem (Information Processing Letters)
- Measured descent: A new embedding method for finite metrics (FOCS 2004 and Geom. Func. Anal.)
- The black-box complexity of nearest neighbor search (ICALP 2004 and Theoretical Computer Science)
- Navigating nets: Simple algorithms for proximity search (SODA 2004)
- Absolute Lipschitz extendability (Comptes Rendus de l'Académie des Sciences)
- Extending Lipschitz functions via random metric partitions (Math. Invent.)
- Metric structures in L_1 : Dimension, snowflakes, and average distortion (Europ. J. of Combinatorics)
- Embedding the diamond graph in L_p and dimension reduction in L_1 (Geom. Func. Anal.)
- Bounded geometries, fractals, and low-distortion embeddings (FOCS 2003)
- The intrinsic dimensionality of graphs (STOC 2003 and Combinatorica)
- Hardness of approximating vertex-connectivity network design problems (SIAM J. on Computing)

Scientific and professional societies of which a member:

- Association for Computing Machinery
- American Mathematical Society

Honors and awards:

- NSF Graduate Research Fellowship
- Institute for Advanced Study Postdoctoral Fellowship
- NSF CAREER Award

Institutional and professional service in last five years:

- Program committees: APPROX 2006 and FOCS 2007
- NSF Computational Geometry review panel

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Levy, Henry

Professor (1983 Ass't, 1988 Assoc, 1994 Full)

Education:

- B.S., Math/Computer Science, Carnegie-Mellon University, 1974
- M.S., Computer Science, University of Washington, 1981

Other Experience:

- Digital Equipment Corporation, 1974 - 1983, Consulting
- Engineer, member of original design team for VMS operating system, system architect for VAXclusters distributed system and VAXstation display systems.

Consulting, patents, etc.:

- U.S. Patent 5,953,543. Environment manipulation for executing modified executables and dynamically-loaded libraries (DLLs) (with T. Romer, A. Wolman, D. Lee, G. Voelker, W. Wong, B. Chen, and B. Bershad). Issued September 14, 1999.
- U.S. Patent 6,014,513. Method for discovering code and data in an x86 binary executable program (with G. Voelker, T. Romer, A. Wolman, D. Lee, W. Wong, B. Chen, and B. Bershad). Issued January 11, 2000.

Principal publications of last five years:

- SpyProxy: Execution-based Detection of Malicious Web Content, with A. Moshchuk, T. Bragin, D. Deville, and S. Gribble. 16th USENIX Security Symposium, August 2007.
- A Safety Oriented Platform for Web Applications, with R. Cox, J. Hansen, and S. Gribble. Proc. of the IEEE Symposium on Security and Privacy, May 2006. (Award Paper).
- A Crawler-Based Study of Spyware on the Web, with A. Moshchuk, T. Bragin, and S. Gribble. Proc. of the 13th Network and Distributed Systems Security Symposium, Feb. 2006.
- Recovering Device Drivers, with M. Swift, M. Annamalai, and B. Bershad. ACM Trans. On Computer Systems 24(4), November 2006. Previous version appeared in Proc. of the 6th Symp. on Operating System Design and Implementation, Dec. 2004. (Award paper.)
- Recovering Device Drivers, with M. Swift, M. Annamalai, and B. Bershad. ACM Trans. On Computer Systems 24(4), November 2006. Previous version appeared in Proc. of the 6th Symp. on Operating System Design and Implementation, Dec. 2004. (Award paper.)
- Semantic Email, with L. McDowell, O. Etzioni, and A. Halevy. Proc. of the 13th International Conference on the World-Wide Web, May 2004 (best paper award runner-up).
- Measurement, Modelling, and Evaluation of a Peer-to-Peer File-Sharing Workload, with K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, and J. Zahorjan. Proc of the 19th ACM Symp. on Operating Systems Principles, Oct. 2003.
- Improving the Reliability of Commodity Operating Systems, with M. Swift and B. Bershad. ACM Trans. on Computer Systems. (Award Paper, 19th ACM Symp. on Operating Systems Principles, Oct. 2003.)
- An Evaluation of Speculative Instruction Execution on Simultaneous Multithreaded Processors, with S. Swanson, L. McDowell, M. Swift, and S. Eggers. ACM Trans. on Computer Systems 21 (3) August 2003.
- Measurement of Internet Content Delivery Mechanisms, with S. Saroiu, K. Gummadi, and S. Gribble. Proc. of the 5th Symposium on Operating Systems Design and Implementation. December 2002. (Award Paper.)

Scientific and professional societies of which a member:

- Fellow, IEEE

- Fellow, Association for Computing Machinery

Honors and awards:

- Wissner-Slivka Chair (endowed chair), 2004
- Microsoft Professor (endowed professorship), 2000
- Fellow of IEEE, 2002
- Fellow of ACM, 1995
- Fulbright Research Scholar Award, 1992
- 14 Award papers
- Distinguished Lectures (UNC/Duke/NC State [2006], Univ. of Illinois [2005], Univ. of Wisconsin [2004], Univ. of British Columbia [2002])

Institutional and professional service in last five years:

- Program Committee:
- 21st ACM Symp. on Operating Systems Principles (2007)
- 11th IEEE/ACM Int. Conf. on Architectural Support for Prog. Lang. and Operating Systems (2004)
- 4th Usenix Symp. on Internet Technologies and Systems (2003)

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - none.

Mones, Barbara

Senior Lecturer (1999 Ass't)

Education:

- BFA, Art Studio and Art History, University of Michigan, 1974
- MFA, Sculpture and Mixed Media, Rhode Island School of Design, 1979
- Post Graduate Certificate, Animation, Sheridan College, 1986

Other Experience:

- Associate Professor (tenured), George Mason University
- Visiting Scholar -- George Washington University
- Animation curriculum design and training, technical writing for research and development area. -- Pacific Data Images
- Animation and technical training and curriculum and professional development -- Industrial Light and Magic
- Web and graphic design -- GLOBE Program, NASA Goddard
- SIGGRAPH 97 Panels Chair, SIGGRAPH Art Show Jury Member, SIGGRAPH Animation Competition Coordinator, SIGGRAPH Education Committee Art Representative
- Member, Visualization And Presentation Committee
- Globe Program, White House, Washington D.C. Professor of Computer Animation And 3D Modeling
- Lugano Academy Of The Electronic Arts, Switzerland

Consulting, patents, etc.:

- Curriculum consulting including Sheridan Animation Program redesign, Marymount College, and several K-12 programs. Consulting at Griffith University, Brisbane, Australia, curriculum review for doctoral program and student reviews of animation production. Consulting for the Museum of the Moving Image, Melbourne, Australia, Consulting for a Phd program in development at the University of Tasmania, Australia,

Principal publications of last five years:

- Artists And Visualization In Education, Eurographics '94
- Is Visualization Struggling under the Myth of Objectivity?, IEEE Visualization '95
- Joan Truckenbrod and Barbara Mones-Hattal, "Interactive Computer Graphics Through Visualization," Interactive Computer Graphics And Telecommunications For Artists And Educators, Edited By Steve Cunningham and Roger Hubbard, Springer Verlag
- Barbara Mones-Hattal and Evans Mandes, "Enhancing Visual Thinking And Learning With Computer Graphics And Virtual Environment Design," Computers And Graphics, Vol. 19, No. 6, Elsevier Science Ltd, U.K.
- Barbara Mones-Hattal, "The Ambiguity Of Dimension," Third International Symposium On The Electronic Arts (Tisea), Sydney, Australia

Scientific and professional societies of which a member:

- ACM SIGGRAPH
- College Art Association
- ASIFA

Honors and awards:

- Group Achievement Award, Globe Visualization Team, National Aeronautics and Space Administration
- Erskine Fellowship, University of Canterbury, Christchurch, New Zealand

Institutional and professional service in last five years:

- Created and edited animation production video for 1999 CSE Graduation (with help from Scott Dakins and UW TV)
- Contributed as needed for several outreach activities related to Animation on campus -- including the annual CSE Open House, the university wide open house, CSE Graduation, the graphics group retreat, Washington Technology Initiative conferences, yearly Affiliates meetings, the yearly Freshman Orientation, the 10 Yr review, and others.
- Recommended and helped implement the move from Odegaard Library to Sieg 327/9 and from 327/9 Sieg to 325/322 Sieg. Responsible for software and equipment procurement for the animation area.
- Completed (directed and produced) eight short animated student films for the department.
- Contributed to several publications related to marketing the computer animation area here including Northwest Magazine, an article in the New York Times, several articles in the Seattle Times Magazine. Provided images for the yearly CSE calendar and various other print media for CSE. Provided DVD's and video for local events such as Bumbershoot. Provided DVD and video of the student work for UWTV which is aired with some frequency. Work with Dxarts (faculty and CSE Representative) on curriculum development, coordinating lecture series events and workshops.

States in which registered: NA

Level of activity: professional societies - high; research - low; consulting/summer work in industry - medium.

Notkin, David

Professor (1984 Ass't, 1989 Assoc, 1994 Full)

Education:

- Sc.B., Computer Science, Brown University, 1977
- Ph.D., Computer Science, Carnegie Mellon University, 1984

Other Experience:

- Lund Institute of Technology, Visiting Researcher, 2006-07.
- IBM Haifa Research Laboratory, Visiting Researcher, 1997-1998.
- Tokyo Institute of Technology, Department of Computer Science, Visiting Associate Professor, Toshiba Endowed Chair of Intellectual Information Systems, 1990.
- Osaka University, Department of Information and Computer Sciences, Visiting Associate Professor, Endowed Chair of Intelligent Information Processing Systems, 1990-1991.

Consulting, patents, etc.:

- Microsoft Research; TRW; Siemens; US Army; Bell Laboratories; Tartan Laboratories; IBM; Xerox PARC; US West/AT; Stoel Rives Boley Jones & Grey; Preston Thorgrimson Shidler Gates & Ellis; MacDonald, Hoague & Bayless; Perkins Coie; i5digital.com (principal scientist, member technical advisory board)

Principal publications of last five years:

- Miryung Kim, David Notkin and Dan Grossman. Automatic Inference of Structural Changes for Matching Across Program Versions. International Conference on Software Engineering (May 2007).
- Tao Xie and David Notkin. Checking Inside the Black Box: Regression Testing By Comparing Value Spectra. IEEE Transactions on Software Engineering SE-31 (10), pp. 869-883 (October 2005).
- Tao Xie and David Notkin. Tool-Assisted Unit-Test Generation and Selection Based on Operational Abstractions. Automated Software Engineering Journal, 2006.
- Tao Xie and David Notkin. Automatically Identifying Special and Common Unit Tests for Object-Oriented Programs. In Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering (November 2005).
- Miryung Kim, Vibha Sazawal, David Notkin and Gail Murphy. An Empirical Study of Code Clone Genealogies. 10th European Software Engineering Conference and the 13th Foundations of Software Engineering (September 2005).
- Tao Xie, Darko Marinov, and David Notkin. Rostra: A Framework for Detecting Redundant Object-Oriented Unit Tests. In Proceedings of the 19th IEEE International Conference on Automated Software Engineering (ASE 2004), Linz, Austria, pp. 196-205, September 2004.
- 9Miryung Kim, Lawrence Bergman, Tessa Lau, and David Notkin. An Ethnographic Study of Copy and Paste Programming Practices in OOPL. International Symposium on Empirical Software Engineering, (August 2004).
- Vibha Sazawal, Miryung Kim, and David Notkin. A Study of Evolution in the Presence of Source-Derived Partial Design Representations. International Workshop on the Principles of Software Evolution (September 2004).
- Jonathan Aldrich, Vibha Sazawal, Craig Chambers, and David Notkin. Language Support for Connector Abstractions. In Proceedings of the European Conference on Object-Oriented Programming (July 2003).
- Jonathan Aldrich, Craig Chambers and David Notkin. Architectural Reasoning in ArchJava. Proceedings of the European Conference on Object-Oriented Programming (June 2002).
- Jonathan Aldrich, Craig Chambers and David Notkin. ArchJava: Connecting Software Architecture to

Implementation. Proceedings of the 2002 International Conference on Software Engineering (May 2002).

Scientific and professional societies of which a member:

- Association for Computing Machinery, Special Interest Groups: Programming Languages, Software Engineering. Institute of Electrical and Electronic Engineers, IEEE Computer Society. Society of the Sigma Xi, Carnegie-Mellon University Chapter, Full Member (1983), Brown University Chapter, Associate Member (1977).

Honors and awards:

- 2007 ACM SIGSOFT Distinguished Service Award
- 2002 R1edu award for Distinguished Faculty Contributions to Online Learning.

Institutional and professional service in last five years:

- Editor in Chief, ACM Transactions on Software Engineering and Methodology (2007-2010)
- Board Member (elected), Computing Research Association (2005-08)
- Chair, Computing Research Association Elections Committee (2006-07)
- Program Co-Chair, 2006 CRA Conference at Snowbird
- Co-Chair, Academic Alliance, National Center for Women in Information Technology (2005-)
- Member, National Advisory Board Committee, Resource Facility for Population Kinetics (2001- , chair 2006-).
- Department Chair (2001-06).

States in which registered: None.

Level of activity: professional societies - high; research - high; consulting/summer work in industry - medium.

Oskin, Mark

Assistant professor (2001 Ass't, 2007 Assoc)

Education:

- B.S., Computer Science, University of California, Davis, 1996
- M.S., Computer Science, University of California, Davis, 2000
- Ph.D., Computer Science, University of California, Davis, 2001

Consulting, patents, etc.:

- Microsoft Research Consultant September 2006 - Present

Principal publications of last five years:

- Martha Mercaldi, Mojtaba Mehrara, Mark Oskin and Todd Austin, Architectural Implications of Brick and Mortar Silicon Manufacturing, International Symposium on Computer Architecture (ISCA) 2007.
- Martha Mercaldi, Steve Swanson, Andrew Petersen, Andrew Putnam, Andrew Schwerin, Mark Oskin and Susan Eggers, Instruction Scheduling for Tiled Dataflow Architectures, Architectural Support for Programming Languages and Operating Systems (ASPLOS) 2006.
- Steve Swanson, Andrew Putnam, Martha Mercaldi, Ken Michelson, Andrew Petersen, Andrew Schwerin, Mark Oskin, and Susan Eggers, Area-Performance Trade-offs in Tiled Dataflow Architectures, International Symposium on Computer Architecture (ISCA) 2006.
- Steven Balensiefer, Lucas Kreger-Stickles, Mark Oskin. An Evaluation Framework and Instruction Set Architecture for Ion-Trap based Quantum Micro-architectures, International Symposium on Computer Architecture (ISCA) 2005
- Steve Swanson, Ken Michelson, Andrew Schwerin and Mark Oskin. WaveScalar In the 36th Annual International Symposium on Microarchitecture (MICRO-36), December 2003
- Mark Oskin, Frederic T. Chong, Isaac Chuang and John Kubiatowicz. Building Quantum Wires: The Long and the Short of it In the 30th Annual International Symposium on Computer Architecture (ISCA), June 2003
- Chris J. Thompson, Sangyun Hahn, and Mark Oskin. Using Modern Graphics Architectures for General-Purpose Computing: A Framework and Analysis International Symposium on Microarchitecture (MICRO), Turkey, Nov. 2002

Scientific and professional societies of which a member:

- ACM

Honors and awards:

- NSF CAREER Award (2001)
- Sloan Foundation Fellowship (2005)

Institutional and professional service in last five years:

- Departmental service (committees, etc.)
- 2002-03 Graduate Student Admissions Committee
- 2003-04 Graduate Student Admissions Committee
- 2004-05 Graduate Student Admissions Committee
- 2005-06 Graduate Student Admissions Committee

- 2006-07 Graduate Student Admissions Committee
- University service (committees, etc.)
- 2006-07 Faculty Senate Representative
- Professional society and other service
- 2003 Program committee member for the International Symposium on Microarchitecture (MICRO)
- 2004 Program committee member for Architectural Support for Programming Languages and Operating Systems (ASPLOS)
- 2004 Program committee member for High Performance Computer Architecture (HPCA)
- 2005 Program committee member for High Performance Computer Architecture (HPCA)
- 2006 Program Chair, ASPLOS Wild and Crazy Ideas V.
- 2006 Co-organizer/Program Chair of WARP: Workshop on Architecture Research Prototyping held in conjunction with ISCA.
- 2007 Program committee member for the International Symposium on Computer Architecture (ISCA)
- 2007 Publications Chair for the International Symposium on Computer Architecture (ISCA)
- National or governmental service
- Dec 2003 NSF Panel Review Member
- May 2004 NSF Panel Review Member
- June 2004 NSF Panel Review Member
- June 2005 NSF Panel Review Member
- Oct 2006 NSF External reviewer (1 proposal)

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - medium.

Perkins, Hal

Lecturer (1998 Ass't)

Education:

- B.S., Mathematics, Arizona State University, 1975
- M.S., Computer Science, Cornell University, 1982

Other Experience:

- Lecturer in Computer Science, Cornell University, 1993-1998.

Scientific and professional societies of which a member:

- ACM (SIGCSE, SIGPLAN, others)
- IEEE & IEEE Computer Society

Honors and awards:

- 1998 Faculty of the Year Award, Cornell University Association of Computer Science Undergraduates

Institutional and professional service in last five years:

- August, 1999, Instructor, Computer Science Track, Microsoft Washington IT Educators Summer Institute, Bellevue Community College
- Panelist, Microsoft Washington IT Educators Summer Institute, Bellevue Community College, August, 2000
- Invited talk, WCERTE spring conference, 2001, WSU Pullman
- Department service: Undergraduate admissions (2000-present), CSE142/143 stewardship (1998-present), FIG Liason (1999-present), Faculty lunch coordination (2000-present)

States in which registered: none

Level of activity: professional societies - low; research - none; consulting/summer work in industry - none.

Popovic, Zoran

Assistant professor (1999 Ass't, 2004 Assoc)

Education:

- B.S., Computer Science, Brown University, 1991
- M.S., Computer Science, Carnegie Mellon University, 1993
- Ph.D., Computer Science, Carnegie Mellon University, 1999

Other Experience:

- Research Associate, University of Washington, Department of CSE (July - September 1999)
- Computer Graphics Researcher, Justsystem Pittsburgh Research Center (summer 1997)
- Research Associate, University of California at Berkeley, Department of Integrative Biology (summer 1996)
- Computer Graphics Software Engineer, Sun Microsystems - Advanced Technology Group (summers of 1990, 1991, 1992)

Consulting, patents, etc.:

- Sportvision Consulting 2007
- Electronic Arts Consulting 2006
- CrowdFlows ver. 1 Non-exclusive trial license issued to Microsoft Corporation, 2006
- CrowdFlows ver. 1 Non-exclusive license issued to Electronic Arts Corporation, 2005
- Style-based IK ver. 1 Non-exclusive license issued to Electronic Arts Corporation, 2003
- Style-based IK ver. 2 Non-exclusive trial license issued to OMG/Vicon, 2004
- Style-based IK ver. 2 Non-exclusive trial license issued to Alias, 2004
- HumanShapes v.1 Non-exclusive trial license issued to Alias, 2004
- HumanShapes v.1 Non-exclusive trial license issued to Discreet, 2004

Principal publications of last five years:

- C. Karen Liu, Aaron Hertzmann, Zoran Popović "Composition of Complex Optimal Multi-Character Motions" in ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2006.
- Brett Allen, Brian Curless, Zoran Popović, Aaron Hertzmann "Learning a correlated model of identity and pose-dependent body-shape variation for real-time synthesis" in ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2006.
- Adrien Treuille, Seth Cooper, Zoran Popović "Continuum Crowds" in ACM Transactions on Graphics (SIGGRAPH) 25(3), 2006.
- Adrien Treuille, Andrew Lewis, Zoran Popović "Model Reduction for Real-time Fluids" in ACM Transactions on Graphics (SIGGRAPH) 25(3), 2006.
- Yeuhi Abe, C. Karen Liu and Zoran Popović, "Momentum-based parameterization of dynamic character motion" in Graphical Models, Volume 68, Issue 2, March 2006, Pages 194-211
- Brown, W.M., Cronk, L., Grochow, K., Jacobson, A., Liu, K., Popović, Z., Trivers, R. "Dance reveals symmetry especially in young men" in Nature, Vol. 438 Num. 7071.
- Steve Capell, Matthew Burkhart, Brian Curless, Tom Duchamp, Zoran Popović "Physically Based Rigging for Deformable Characters" in ACM SIGGRAPH / Eurographics Symposium on Computer Animation, 2005.
- C. Karen Liu, Aaron Hertzmann, Zoran Popović "Learning Physics-based Motion Style with Inverse Optimization," in ACM Transactions on Graphics (SIGGRAPH) 24(3), pp. 1071-1081, 2005.
- Yeuhi Abe, C. Karen Liu, Zoran Popović "Momentum-based Parameterization of Dynamic Character Motion," in ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2004.

- Keith Grochow, Steve L. Martin, Aaron Hertzmann, Zoran Popović “Style-Based Inverse Kinematics,” in ACM Transactions on Graphics (SIGGRAPH 2004).
- Antoine McNamara, Adrien Treuille, Zoran Popović, Jos Stam, “Fluid Control using the Adjoint Method,” in ACM Transactions on Graphics (SIGGRAPH) 2004.
- Brett Allen, Brian Curless, Zoran Popović, “Exploring the space of human body shapes: data-driven synthesis under anthropometric control,” in Proceedings of SAE Digital Human Modeling for Design and Engineering Conference 2004.
- C. Karen Liu and Zoran Popović, “Synthesis of Complex Dynamic Motion from Simple Animations”, in ACM Transactions on Graphics (SIGGRAPH) 2002.
- Brett Allen, Brian Curless, and Zoran Popović, “Articulated Body Deformation from Range Scan Data”, in ACM Transactions on Graphics (SIGGRAPH) 2002.
- Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popović, “Skeletal Animation of Deformable Characters”, in ACM Transactions on Graphics (SIGGRAPH) 2002.
- Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popović, “A Multiresolution Framework for Elastic Deformations”, in ACM SIGGRAPH Symposium on Computer Animation 2002.

Scientific and professional societies of which a member:

- ACM

Honors and awards:

- 2004 ACM SIGGRAPH Significant New Researcher Award
- 2003 Alfred P. Sloan Fellowship
- 2001 NSF CAREER Award
- 1995-99 Schlumberger Research Fellowship,
- 1990-91 Baxter-Travenol Scholarship (merit-based full-tuition scholarship)

Institutional and professional service in last five years:

- 2007 SIGGRAPH 2007. Program Committee.
- 2006 SIGGRAPH 2006. Program Committee.
- 2005 MIRAGE 2005. Program Committee.
- 2004 ACM SIGGRAPH/Eurographics Symposium for Computer Animation 2004. Program Committee.
- 2004 Eurographics 2004. Topic Chair.
- 2003 SIGGRAPH 2003. Program Committee.
- 2002 Pacific Graphics 2002. Program Committee.
- 2002 ACM SIGGRAPH Symposium on Computer Animation 2002. Program Committee.

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Rao, Rajesh

Associate professor (2000 Ass't)

Education:

- Ph.D., Computer Science, University of Rochester, 1998
- M.S., Computer Science, University of Rochester, 1994
- B.S., Computer Science and Mathematics, Angelo State University, 1992

Principal publications of last five years:

- Probabilistic Models of the Brain: Perception and Neural Function, Rajesh P. N. Rao, Bruno A. Olshausen and Michael S. Lewicki (Eds.), Cambridge, MA: MIT Press, 2002.
- The Bayesian Brain: Probabilistic Approaches to Neural Coding, Kenji Doya, Shin Ishii, Alexandre Pouget, and Rajesh P. N. Rao (Eds.), Cambridge, MA: MIT Press, 2007.
- Kai J. Miller, Eric C. Leuthardt, Gerwin Schalk, Rajesh P. N. Rao, et al. "Spectral Changes in Cortical Surface Potentials during Motor Movement", The Journal of Neuroscience (to appear), 2007.
- Pradeep Shenoy, Kai J. Miller, Jeffrey G. Ojemann, and Rajesh P. N. Rao. "Finger Movement Classification for an Electrographic BCI," The 3rd International IEEE EMBS Conference on Neural Engineering, 2007.
- Christian J. Bell, Pradeep Shenoy, Rawichote Chalodhorn, and Rajesh P. N. Rao. "An Image-based Brain-Computer Interface Using the P3 Response," The 3rd International IEEE EMBS Conference on Neural Engineering, 2007.
- Pradeep Shenoy, Matthias Krauledat, Benjamin Blankertz, Rajesh P. N. Rao and Klaus-Robert Mueller. "Towards adaptive classification for BCI" The Journal of Neural Engineering, Vol. 3(1), pp. R13-23, 2006.
- Eric C. Leuthardt, Kai J. Miller, Gerwin Schalk, Rajesh P. N. Rao, and Jeffrey G. Ojemann. "Electrographic-based Brain Computer interface - the Seattle experience" IEEE Trans Neural Syst Rehab Eng, Vol. 14(2), pp. 194-198, 2006.
- Rajesh P. N. Rao. "Neural models of Bayesian belief propagation," in The Bayesian Brain: Probabilistic Approaches to Neural Coding, Doya K, Ishii S, Pouget A, Rao RPN (Eds.), Cambridge, MA: MIT Press, 2007.
- Matt Hoffman, David B. Grimes, Aaron P. Shon, and Rajesh P. N. Rao. "A Probabilistic Model of Gaze Imitation and Shared Attention," Neural Networks, Vol. 19(3), 299-310, 2006.
- Deepak Verma and Rajesh P. N. Rao. "Goal-Based Imitation as Probabilistic Inference over Graphical Models," Advances in Neural Information Processing Systems 18, Cambridge, MA: MIT Press, 1393-1400, 2006.
- Rajesh P. N. Rao. "Bayesian Inference and Attentional Modulation in the Visual Cortex," Neuroreport, Vol. 16(16), 1843-1848, 2005.
- K. J. Miller, G. Schalk, J. W. Miller, R. P. N. Rao, E. C. Leuthardt, J. M. Zacks, and J. G. Ojemann. "Selective attention effects associated with very high frequency changes in human primary visual cortex," Soc Neurosci Abs, 2005.
- Rajesh P. N. Rao. "Hierarchical Bayesian Inference in Networks of Spiking Neurons," Advances in Neural Information Processing Systems 17, Cambridge, MA: MIT Press, pp. 1113-1120, 2005.
- Pradeep Shenoy and Rajesh P. N. Rao. "Dynamic Bayesian Networks for Brain-Computer Interfaces," Advances in Neural Information Processing Systems 17, Cambridge, MA: MIT Press, pp. 1265-1272, 2005.
- David B. Grimes and Rajesh P. N. Rao. "Sparse Bilinear Models for Invariant Vision," Neural Computation, Vol. 17(1), pp. 47-73, 2005.
- Rajesh P. N. Rao, Aaron P. Shon, and Andrew N. Meltzoff. "A Bayesian Model of Imitation in Infants and Robots," in Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions, K. Dautenhahn and C. L. Nehaniv (eds.), Cambridge University Press, UK, 2007.
- Chris L. Baker, Aaron P. Shon, and Rajesh P. N. Rao. "Learning Temporal Clusters with Synaptic

Facilitation and Lateral Inhibition," Neurocomputing, Vol. 65-66, 2005.

- Aaron P. Shon, David B. Grimes, Chris L. Baker, and Rajesh P. N. Rao. "A Probabilistic Framework for Model-Based Imitation Learning," Proc. of the 26th Annual Meeting of the Cognitive Science Society, 2004.
- Rajesh P. N. Rao and Terrence J. Sejnowski. "Self-Organizing Neural Systems based on Predictive Learning" (Invited Paper) Philosophical Transactions of the Royal Society: Mathematical, Physical and Engineering Sciences (Proceedings of the Nobel Symposium on Self-Organization), Vol. 361(1807), 2003.
- Rajesh P. N. Rao and Terrence J. Sejnowski. "Predictive Coding, Cortical Feedback, and Spike-Timing Dependent Plasticity" in Probabilistic Models of the Brain: Perception and Neural Function, R. P. N. Rao, B. A. Olshausen and M. S. Lewicki (Eds.), Cambridge, MA: MIT Press, pp. 297-315, 2002.
- Dana H. Ballard, Zuohua Zhang, and Rajesh P. N. Rao. "Distributed Synchrony: A Probabilistic Model of Neural Signaling" in Probabilistic Models of the Brain: Perception and Neural Function, R. P. N. Rao, B. A. Olshausen and M. S. Lewicki (Eds.), Cambridge, MA: MIT Press, pp. 273-283, 2002.

Scientific and professional societies of which a member:

- Society for Neuroscience, IEEE, ACM

Honors and awards:

- ONR Young Investigator Award, 2003-2006.
- David and Lucile Packard Fellowship, 2002-2007.
- NSF CAREER Award, 2002-2007.
- Alfred P. Sloan Research Fellowship, 2001-2003.
- Alfred P. Sloan Postdoctoral Fellowship, Salk Institute for Biological Studies, 1997-2000.
- Presidential Fellowship for Graduate Studies, State University of New York, Buffalo, 1992 (declined in favor of Univ. of Rochester Graduate Assistantship).

Institutional and professional service in last five years:

- CSE Executive Committee, 2006-2007.
- CSE SpaceManagement Committee, 2004-present (with Prof. Paul Beame, Erik Lundberg, Tracy Bartholomew, Chris Cunningham).
- CSE Course Scheduling Committee, 2004 (with Prof. Richard Anderson).
- UNIVERSITY SERVICE Faculty Field Tour, 2002. Participated in a five-day faculty bus tour around the state of Washington with President Richard L. McCormick to further relations of UW with the residents of Washington state.
- Mentor, UW Early Identification Program (EIP), 2002. Goal: To encourage and assist UW undergraduate students from underrepresented, educationally, and economically disadvantaged groups to enter graduate school. Advisees: Chris Baker and Ie Ming Tjam.

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Reges, Stuart

Lecturer (2004 Ass't)

Education:

- M.S., Computer Science, Stanford University, 1982
- B.S., Mathematics, Case Western Reserve University, 1979

Other Experience:

- 5/01-6/04, Associate Head for Undergraduate Studies, Department of Computer Science, University of Arizona.
- 8/96-6/04, Senior Lecturer, Department of Computer Science, University of Arizona.
- 9/88-5/91, Senior Lecturer, Department of Computer Science, Stanford University.
- 6/85-12/89, Chief Reader, Computer Science Advanced Placement Exam, Educational Testing Service.
- 9/85-3/88, Assistant Chairman for Education, Department of Computer Science, Stanford University.

Principal publications of last five years:

- Building Java Programs, 2007, Addison Wesley, coauthored with Marty Stepp
- "Back to Basics in CS1 and CS2", Thirty-seventh SIGCSE Technical Symposium on Computer Science Education (2006)
- "Using Undergraduates as Teaching Assistants at a State University", Thirty-fourth SIGCSE Technical Symposium on Computer Science Education (2003)
- "Can C# Replace Java in CS1 and CS2?", Seventh ITICSE Conference (2002)
- "Teaching tips we wish they'd told us before we started", Thirty-eighth SIGCSE Technical Symposium on Computer Science Education (2007)
- "Nifty Assignments: Anagrams", Thirty-seventh SIGCSE Technical Symposium on Computer Science Education (2006)
- "Resolved: Objects Early has Failed", Thirty-sixth SIGCSE Technical Symposium on Computer Science Education (2005)
- "Nifty Assignments: Lunar Lander", Thirty-fourth SIGCSE Technical Symposium on Computer Science Education (2003)

Scientific and professional societies of which a member:

- ACM

Honors and awards:

- Honorable Mention, Student ACM Teaching Award (UW, 2006)
- Honors College Outstanding Advisor (U of A, 2002)
- College of Science Distinguished Teaching Award (U of A, 1998)
- Honorable Mention, Associated Students of Stanford University Teaching Award (Stanford, 1991)
- School of Engineering Distinguished Advisor Award (Stanford, 1986)
- Dinkelspiel Award for Outstanding Service to Undergraduate Education (Stanford, 1985)

Institutional and professional service in last five years:

- Undergrad Admissions Committee (2004-2007)

- 142/143 Stewardship Committee (2004-2007)

States in which registered: none

Level of activity: professional societies - low; research - none; consulting/summer work in industry - none.

Ruzzo, Walter L.

Professor (1977 Ass't, 1982 Assoc, 1990 Full)

Education:

- B.S., Mathematics, Caltech, 1968
- Ph.D., Computer Science, UC Berkeley, 1978

Other Experience:

- 5 years industrial experience, 1968-1973

Principal publications of last five years:

- D.J. Patterson, K. Yasuhara and W.L. Ruzzo, "Pre-mRNA Secondary Structure Prediction Aids Splice Site Prediction," Pacific Symposium on Biocomputing 2002, 223-234. World Scientific (Altman, et al., eds.).
- M.T. Barrett, K.Y. Yeung, W.L. Ruzzo, L. Hsu, P.L. Blount, R. Sullivan, H. Zarbl, J. Delrow, P.S. Rabinovitch, and B.J. Reid, "Transcriptional Analyses of Barrett's Metaplasia and Normal Upper GI Mucosae," *Neoplasia*, 4 (2) 121-128, 2002.
- J. Jaeger, R. Sengupta and W.L. Ruzzo, "Improved Gene Selection For Classification Of Microarrays," Pacific Symposium on Biocomputing 2003, 53-64.
- Z. Weinberg and W.L. Ruzzo, "Faster Genome Annotation of Non-coding RNA Families Without Loss of Accuracy" 8th Int Conf Research in Comp Mol Biol (RECOMB 2004), pp 243-251, March 2004, San Diego, CA.
- Z. Weinberg and W.L. Ruzzo, "Exploiting Conserved Structure for Faster Annotation of Non-coding RNAs Without Loss of Accuracy." 12th Int Conf Intell Syst Mol Biol (ISMB 2004), and Bioinformatics, 20 (suppl_1) i334-i341, 2004.
- E. Mulvihill, J. Jaeger, R. Sengupta, W.L. Ruzzo, C. Reimer, S. Lukito and S. M. Schwartz, "Atherosclerotic Plaque Smooth Muscle Cells Have a Distinct Phenotype," *Arterioscler Thromb Vasc Biol*, 24(7):1283-9, July 2004.
- M. Mandal, M. Lee, J.E. Barrick, Z. Weinberg, G.M. Emilsson, W.L. Ruzzo, R.R. Breaker, "A Glycine-dependent Riboswitch that Uses Cooperative Binding to Control Gene Expression in Bacteria." *Science*, 306:275-279, 8 Oct 2004.
- J.E. Barrick, N. Sudarsan, Z. Weinberg, W.L. Ruzzo and R.R. Breaker, "6S RNA is a widespread regulator of eubacterial RNA polymerase that resembles an open promoter," *RNA*, 11(5):774-84 May, 2005.
- P.C.H. Hsieh, R.D. Kenagy, E.R. Mulvihill, J.P. Jeanette, X. Wang, C.M.C. Chang, Z. Yao, W.L. Ruzzo, S. Justice, K.L. Hudkins, C.E. Alpers, S. Berceli, A.W. Clowes, "BMP4: Potential Regulator of Shear Stress-induced Graft Neointimal Atrophy." *Journal of Vascular Surgery*, 43(1), 150-158, Jan 2006.
- Z. Weinberg and W.L. Ruzzo, "Sequence-based heuristics for faster annotation of non-coding RNA families," *Bioinformatics*, 22(1):35-39, Jan 2006.
- Z. Yao, Z. Weinberg and W.L. Ruzzo, "CMfinder—A Covariance Model Based RNA Motif Finding Algorithm," *Bioinformatics*. 44(4): 445-452, 15 February 2006.
- Z. Yao and W.L. Ruzzo, "A regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data." *BMC Bioinformatics* 7(Suppl 1):S11, 20 March 2006.
- J.A. Eisen, R.S. Coyne, M. Wu, D. Wu, M. Thiagarajan, J.R. Wortman, J.H. Badger, Q. Ren, P. Amedeo, K.M. Jones, L.J. Tallon, A.L. Delcher, S.L. Salzberg, J.C. Silva, B.J. Haas, W.H. Majoros, M. Farzad, J.M. Carlton, R.K. Smith Jr., J. Garg, R.E. Pearlman, K.M. Karrer, L. Sun, G. Manning, N.C. Elde, A.P. Turkewitz, D.J. Asai, D.E. Wilkes, Y. Wang, H. Cai, K. Collins, B.A. Stewart, S.R. Lee, K. Wilamowska, Z. Weinberg, W.L. Ruzzo, D. Wloga, J. Gaertig, J. Frankel, C.-C. Tsao, M.A. Gorovsky, P.J. Keeling, R.F. Waller, N.J. Patron, J.M. Cherry, N.A. Stover, C.J. Krieger, C. del Toro, H.F. Ryder, S.C. Williamson, R.A. Barbeau, E.P. Hamilton, E. Orias, "Macronuclear Genome Sequence of the Ciliate Tetrahymena

thermophila, a Model Eukaryote." PLoS Biol 4(9): e286, Sep 2006.

Institutional and professional service in last five years:

- Department
- Grad Admissions (2003--04)
- Qualls (2003--07, Chair)
- Non-majors Course Coordinator (w/ Perkins) 2006--
- College
- Faculty Council (2001--02)
- Faculty Council (2003--04)
- Faculty Council (2006)
- Council on Educational Policy (2007--)
- University
- Computational Molecular Biology Program Advisory Board, inception to present.
- Genome Training Grant Advisory Committee, Spring 2005 -- present
- Fred Hutchinson Cancer Research Center Faculty Search
- Committee for Computational Biology, Fall 2005--Spring 2006.
- Professional
- NSF Panels
- Theory of Computing Panel (Algorithms), March 6--8, 2002.
- EMT Panel, March 2005.
- EMT Panel, March 2006.
- Program Committees
- 2003 IEEE Computer Society Bioinformatics Conference
- 2005 RECOMB 05: 9th Annual International Conference on Computational Molecular Biology
- 2005 BSB 05: Brazilian Symposium on Bioinformatics, July, 2005, Sao Leopoldo, Brazil.
- 2006 APBC 06: Fourth Asia-Pacific Bioinformatics Conference, Taipei, Taiwan, Feb. 13--16, 2006.

States in which registered: none

Level of activity: professional societies - none; research - high; consulting/summer work in industry - none.

Salesin, David

Associate professor (1992 Ass't, 1996 Assoc)

Education:

- Ph.D., Computer Science, Stanford University, 1991
- S.B., Computer Science, Brown University, 1983

Other Experience:

- Cornell University, Program of Computer Graphics. Visiting Assistant Professor, 1991-92.
- Digital Equipment Corporation. Student intern at Paris Research Lab, 1988; at Systems Research Center, summers 1989-90.
- Pixar. Computer graphics researcher, summer 1987.
- Stanford University. Research assistant, 1986-88; teaching fellow in computer graphics, 1988.
- Sogitec Audiovisuel, Paris. Computer graphics consultant, 1986.
- Lucasfilm, Ltd. Computer graphics researcher, 1983-85.

Consulting, patents, etc.:

- Method and system for generating floral ornament, with M. Wong and D. Zongker. Filed 1998.
- Method and system for multiresolution video, with A. Finkelstein and C. Jacobs. Filed 1997.
- Method and system for virtual cinematography, with M. Cohen and L. He. Filed 1996.

Principal publications of last five years:

- Wavelets for Computer Graphics: Theory and Applications, with E. Stollnitz and T. DeRose. Morgan-Kaufmann Publishers, Inc., San Francisco, 1996. ISBN 1-55860-375-1.
- SIGGRAPH papers
- Escherization, with C. Kaplan. Proceedings of SIGGRAPH 2000, in Computer Graphics Proceedings, Annual Conference Series, July 2000. To appear.
- Surface light fields for 3D photography, with D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T.
- Video textures, with A. Shödl, R. Szeliski, and I. Essa. Proceedings of SIGGRAPH 2000, in Computer Graphics Proceedings, Annual Conference Series, July 2000. To appear.
- Environment matting and compositing, with D. Zongker, D. Werner, and B. Curless. Proceedings of SIGGRAPH 99, in Computer Graphics Proceedings, Annual Conference Series, 205-214, August 1999.
- Reproducing color images using custom inks, with E. Stollnitz and V. Ostromoukhov. Proceedings of SIGGRAPH 98, in Computer Graphics Proceedings, Annual Conference Series, 267-274, July 1998.
- Computer-generated floral ornament, with M. Wong, D. Zongker. Proceedings of SIGGRAPH 98, in Computer Graphics Proceedings, Annual Conference Series, 423-434, July 1998.
- Multiperspective panoramas for cel animation, with D. Wood, A. Finkelstein, J. Hughes, and C. Thayer. Proceedings of SIGGRAPH 97, in Computer Graphics Proceedings, Annual Conference Series, 243-250, August 1997.
- Orientable textures for image-based pen-and-ink illustration, with M. Salisbury, M. Wong, and J. F. Hughes. Proceedings of SIGGRAPH 97, in Computer Graphics Proceedings, Annual Conference Series, 401-406, August 1997. Also available as Department of Computer Science and Engineering Technical Report TR 97-01-01, University of Washington, 1997.
- Computer-generated watercolor, with C. J. Curtis, S. E. Anderson, J. E. Seims, and K. W. Fleischer. Proceedings of SIGGRAPH 97, in Computer Graphics Proceedings, Annual Conference Series, 421-430, August 1997.
- Scale-dependent reproduction of pen-and-ink illustrations, with M. Salisbury, C. Anderson, and D. Lischinski. Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series,

461-468, August 1996. Also available as Department of Computer Science and Engineering Technical Report TR 96-01-02, University of Washington, 1996.

- Rendering parametric surfaces in pen and ink, with G. Winkenbach. Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series, 469-476, August 1996. Also available as Department of Computer Science and Engineering Technical Report TR 96-01-05, University of Washington, 1996.
- Fast multiresolution image querying, with C. E. Jacobs and A. Finkelstein. Proceedings of SIGGRAPH 95, in Computer Graphics Proceedings, Annual Conference Series, 277-286, August 1995. Reprinted in Investing in the Future: 1946-1996, commemorating the 50th anniversary of the Office of Naval Research, 48-57, 1996. Also available as Department of Computer Science and Engineering Technical Report TR 95-01-06, University of Washington, 1995.

Scientific and professional societies of which a member:

- Association for Computing Machinery
- Institute of Electrical and Electronics Engineers
- Eurographics

Honors and awards:

- The Carnegie Foundation for the Advancement of Teaching and the Council for the Advancement and Support of Education 1998-1999
- Washington Professor of the Year Award.
- University of Washington Distinguished Teaching Award, 1997.
- University of Washington Award for Outstanding Faculty Achievement in the College of Engineering, 1996.
- Rensselaer Polytechnic Institute Mathematics and Science Award, 1978.
- Bausch and Lomb Science Award, 1978.

Institutional and professional service in last five years:

- Review committees and editorial boards:
- ISAMA 2000 (The Second Annual Conference of the International Society of the Arts, Mathematics and Architecture). Conference Director, with N. Friedman, 2000.
- NPAR 2000 (First Annual Symposium on Non Photorealistic Animation and Rendering). Conference co-chair, with J.-D. Fekete, 2000.
- SIGGRAPH Papers Committee, 1995, 1996, 1998, 2000.
- IEEE Transactions on Visualization and Computer Graphics. Editorial board member, since 1999.
- The Journal of Graphics Tools. Founder and Consulting Editor, since 1995.
- National Science Foundation, CAREER Awards Review Panel, 1999.
- National Science Foundation Science & Technology Center for Computer Graphics and Scientific Visualization, Director's Pool Review Committee, 1996, 1997, 1998.

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - high.

Seitz, Steven

Assistant professor (2000 Ass't, 2003 Assoc)

Education:

- Ph.D., Computer Science, University of Wisconsin, 1997
- B.A., Computer Science, Mathematics, University of California, Berkeley, 1991

Other Experience:

- Assistant Professor of Robotics and Computer Science, Carnegie Mellon University, October 1997 - July 2000
- Adjunct Assistant Professor, Carnegie Mellon University, July 2000 - present
- Postdoctoral Researcher, Microsoft Research, October 1997 - August 1998.

Consulting, patents, etc.:

- Consultant, Vision Technology Group, Microsoft Research (since 1997, ongoing)
- "Photorealistic scene reconstruction by voxel coloring", with C. R. Dyer, U.S. Patent No. 6,363,170, March 26, 2002
- 2. "Method and system for obtaining visual information from an image sequence using visual tunnel analysis", with S. B. Kang, P. J. Sloan, U.S. Patent No. 6,642,924, November 4, 2003

Principal publications of last five years:

- Journals:
- L. Zhang and S. M. Seitz, Estimating optimal parameters for MRF stereo from a single image pair, IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), vol. 29, no. 2, 2007, pp. 331-342.
- N. Snavely, S. M. Seitz, and R. Szeliski, Photo tourism: exploring photo collections in 3D. in ACM Trans. on Computer Graphics (SIGGRAPH Proceedings), vol. 25, no. 3, 2006, pp. 835-846.
- D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz, Schematic storyboarding for video visualization and editing. ACM Trans. on Computer Graphics (SIGGRAPH Proceedings), vol. 25, no. 3, 2006, pp. 862-871.
- A. Hertzmann and S. M. Seitz. Example-based photometric stereo: shape reconstruction with general, varying BRDFs, IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), vol. 27, no. 8, 2005, pp. 1254-1264.
- L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime Faces: High-resolution capture for modeling and animation, in ACM Trans on Computer Graphics (SIGGRAPH Proceedings), vol. 23, no. 3, 2004, pp. 548-558.
- S. M. Seitz, A. Kalai, and H. Shum, Omnivergent stereo, International Journal of Computer Vision, 2002, vol. 48, no. 3, pp. 159-172.
- S. M. Seitz and K. N. Kutulakos, Plenoptic image editing, International Journal of Computer Vision, 2002, vol. 48, no. 2, pp. 115-129.
- S. M. Seitz and J. Kim, The space of all stereo images, International Journal of Computer Vision, Marr Prize Special Issue, 2002, vol. 48, no. 1, pp. 21-38
- F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun, EM, MCMC, and chain flipping for structure from motion with unknown correspondence, Machine Learning, special issue on Markov chain Monte Carlo methods, 2003, vol. 50, pp. 45-71
- Refereed Conference Papers:
- S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms, Proc. Computer Vision and Pattern Recognition Conf., 2006, pp. 519-526.

- M. Goesele, S. M. Seitz and B. Curless. Multi-View Stereo Revisited, Proc. Computer Vision and Pattern Recognition Conf., 2006, pp. 2402-2409.
- D. B. Goldman, B. Curless, A. Hertzmann and S. M. Seitz. Shape and spatially-varying BRDFs from photometric stereo, Proc. International Conference on Computer Vision, 2005, pp. 341-348.
- S. M. Seitz, Y. Matsushita and K. N. Kutulakos. A theory of inverse light transport, Proc. International Conference on Computer Vision, 2005, pp. 1440-1447.
- A. Hertzmann, N. Oliver, B. Curless, and S. M. Seitz, Curve analogies, 13th Euro-graphics Workshop on Rendering, 2002, pp. 233-245
- K. S. Bhat, S. M. Seitz, J. Popovic, and P. K. Khosla, Computing the physical parameters of rigid-body motion from video, Proc. European Conference on Computer Vision, 2002, pp. 551-566
- L. Zhang, B. Curless, and S. M. Seitz, Rapid shape acquisition using color structured light and multi-pass dynamic programming, Proc. Symposium on 3D Data Processing Visualization and Transmission (3DPVT), 2002

Scientific and professional societies of which a member:

- Senior member IEEE
- member ACM

Honors and awards:

- Short-Dooley Career Development Associate Professorship, 2005
- Alfred P. Sloan Fellowship, 2002
- Office of Naval Research Young Investigator Award, 2002
- David Marr Prize, for the best paper at the 8th International Conference on Computer Vision, 2001
- NSF CAREER AWARD, May 2000
- DAVID MARR PRIZE, for the best paper at the International Conference on Computer Vision,

Institutional and professional service in last five years:

- Engineering Writing Assessment (University of Washington)
- Robotics Minor Committee (Carnegie Mellon University)
- Faculty Recruiting Committee (Carnegie Mellon University)
- Associate Editor, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001-2006
- Editorial Board, The Visual Computer, International Journal of Computer Graphics, Springer, 2000-2005
- Editorial Board, Graphical Models, Academic Press, 2002-2005
- Area Chair, CVPR (Computer Vision and Pattern Recognition Conf.), 2007
- Area Chair, ICCV (International Conf. on Computer Vision), 2005
- SIGGRAPH, 2005
- Area Chair, CVPR, 2004
- Area Chair, ECCV (European Conf. On Computer Vision), 2004

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - medium.

Shapiro, Linda

Professor (1974 Ass't, 1981 Assoc, 1989 Full)

Education:

- Ph.D., Computer Science, University of Iowa, 1974
- M.S., Computer Science, University of Iowa, 1972
- B.S., Mathematics, University of Illinois, 1970

Other Experience:

- Director of Intelligent Systems, Machine Vision International, Ann Arbor, Michigan, 1984-1986.

Consulting, patents, etc.:

- Consulting for U.S. Dept of Education
- Consulting for CIA Office of Research and Development

Principal publications of last five years:

- Pan, S., G. Shavit, M. Penas-Centeno, D.-H. Xu, L. Shapiro, R. Ladner, E. Riskin, W. Hol, D. Meldrum, "Automated Classification of Protein Crystallization Images Using Support Vector Machines with Scale-Invariant Texture and Gabor Features," Biological Crystallography, Vol. D62, 2006.
- Ruiz-Correa, S., L. G. Shapiro, M. Meila, G. Berson, M. L. Cunningham, and R. W. Sze,
- "Symbolic Signatures for Deformable Shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 1, 2006, pp. 75-90.
- Shapiro, L. G., E. Chung, L. Detwiler, J. Mejino, Jr., A. Agoncillo, J. F. Brinkley, and C. Rosse, "Processes and Problems in the Formative Evaluation of an Interface to the Foundational Model of Anatomy Knowledge Base," Journal of the American Medical Informatics Association, Vol. 12, 2005, pp. 35-46.
- Ye, M., R. M. Haralick, and L. G. Shapiro, "Estimating Piecewise-Smooth Optical Flow with Global Matching and Graduated Optimization," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 12, 2003, pp. 1625-1630.
- Chou, Y. and L. G. Shapiro, "A Hierarchical Multiple-Classifer Learning Algorithm," Pattern Analysis and Applications, Vol. 6, 2003, pp. 150-168.
- Hinshaw, K.P., Poliakov, A.V., Moore, E.B., Martin, R.F., Shapiro, L.G. and Brinkley, J.F. "Shape-Based Cortical Surface Segmentation for Visualization Brain Mapping," Neuroimage, Vol. 16, 2002, pp. 295-316.

Scientific and professional societies of which a member:

- Institute of Electrical and Electronics Engineers
- American Association for Artificial Intelligence
- Pattern Recognition Society

Honors and awards:

- Phi Beta Kappa, 1969
- Pattern Recognition Society Best Paper, 1984, 1989, and 1995
- Pattern Recognition Society Honorable Mention, 1985, 1987, and 1999
- Fellow of the IEEE, 1996

- Fellow of the IAPR, 2000

Institutional and professional service in last five years:

- Editorial Board Member, Pattern Recognition, beginning March 1986 and ongoing.
- Editorial Board Member, Computer Vision and Image Understanding, ongoing.
- Editorial Board Member, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993- 1996.
- Program Committee, IEEE Workshop on Content-Based Access of Image and Video Libraries, 1997, 1998, 1999
- Program Committee, Computer Vision and Pattern Recognition, 1999
- Area Chair, Program Committee, Computer Vision and Pattern Recognition, 1998
- Director, Computer Systems Group, Washington Technology Center, 1992-98
- Co-Chair, ICPR2002 Track on Biomedical and Multimedia Applications
- Co-Chair, IEEE Conference on Computer Vision and Pattern Recognition, 2008

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Snyder, Lawrence

Professor (1983 Ass't)

Education:

- B.A., Mathematics and Economics, University of Iowa, 1968
- Ph.D., Computer Science, Carnegie Mellon University, 1973

Other Experience:

- Assistant/Associate Professor, Yale University, 1973-1980
- Associate/Professor, Purdue, 1980-1983
- Visiting Scholar, University of Washington, 1979-80
- Visiting Scholar, MIT and Harvard, 1987-88
- Visiting Professor, University of Sydney, 1994-95

Consulting, patents, etc.:

- Chaos Router System, US Patent No: 5,151,900, September 29, 1992

Principal publications of last five years:

- A Programmer's Guide to ZPL, MIT Press, 1999
- Being Fluent With Information Technology, National Academy Press, 1999
- A Comparison of Input and Output Driven Routers, with M. Fulgham, Proc. 2nd International EuroPar Conference, LNCS 1123, Springer Verlag pp. 195-204, 1996
- Ironman: A Machine INdependent Parallel Communication Abstraction to Replace Message Passing in Compilers, with S-E Choi and B. Chamberlain, Languages and Compilers for Parallel Computing, Springer Verlag, pp. 361-375, 1996
- Portable Performance of Data Parallel Languages, with T. Ngo and B. Chamberlain Proc. SC97: High Performance Networking and Computing, 1997
- Triplex: Multiclass Routing, with M. Fulgham, ACM Symposium Parallel Algorithms and Architectures, pp. 127-138, 1997
- Integrated Multi-class Routing, with M. Fulgham, 2nd Parallel Computer Routing and Communication Workshop, Springer Verlag, 1997
- The Case for Chaotic Adaptive Routing, with K. Bolding and M. Fulgham, IEEE Transactions on Computers, 46(12):1281-1291, 1997
- ChaosLAN: Design and Implementation of a Gigabit LAN Using Chaotic Routing, with N. McKenzie, K. Bolding and C. Ebeling, 2nd Parallel Computer Routing and Communication Workshop, Springer Verlag, pp. 211-223, 1997
- Quantifying the Effect5s of Communicatoion Optimizations, with S-E Choi, Int'l Conference on Parallel Processing, pp. 218-222, 1997
- Factor-Join: A Unique Approach to Compiling Array Languages for Parallel Machines, with B. Chamberlain, S-E Choi, E. Lewis, C. Lin and W. Weathersby, Languages and Compilers for Parallel Computing, Springer Verlag, pp. 481-500, 1997
- A Compiler Abstraction for Machine Independent Parallel Communication Generation, with B. Chamberlain and S-E Choi, Languages and Compilers for Parallel Computers, Springer Verlag, pp. 261-276, 1997
- Abstractions for Portable, Scalable Parallel Programming, with G. Alverson, W. Griswold, C. Lin and D. Notkin, IEEE Transactions on Parallel and Distributed Systems, 9(1):71-86, 1998
- ZPL's WYSIWYG Performance Model, with B. Chamberlain, S-E Choi, E Lewis, C. Lin and W. Weathersby, IEEE Workshop on High Level Parallel Programming Models, pp. 50-61, 1998

- The Implementation and Evaluation of Fusion and Contraction in Array Languages, with E. Lewis and C. Lin, ACM Programming Languages and Implementation Symposium, pp. 50-59, 1998
- The Case for High-Level Parallel Programming in ZPL, with B. Chamberlain, S-E Choi, E. Lewis, C. Lin and W. Weathersby, IEEE Computational Science and Engineering, 5(3):76-86, 1998
- Problem Space Promotion and Its Evaluation as a Technique for Efficient Parallel Computation, with B. Chamberlain and E. Lewis, 13th Int'l Conference on Supercomputing, pp. 311-318, 1999
- Regions: An Abstraction for Expressing Array Computation, with B. Chamberlain, E. Lewis and C. Lin, ACM International Conference on Array Languages, pp. 41-49, 1999
- Array Language Support for Wavefront and Pipelined Computations, with B. Chamberlain and E. Lewis, Workshop on Languages and Compilers for Parallel Computing, 1999
- ZPL: A Machine Independent Language for Parallel Computers, with B. Chamberlain, S-E Choi, E. Lewis, C. Lin and W. Weathersby, IEEE Transactions on Software Engineering, 2000 (to appear)
- Pipelining Wavefront Computations: Experiences and Performance, with E. Lewis, IEEE Workshop on High Level Parallel Programming Models, pp. 261-268, 2000

Scientific and professional societies of which a member:

- ACM -- Association for Computing Machinery
- IEEE -- Institute for Electrical and Electronic Engineers
- SIAM -- Society of Industrial and Applied Mathematicians

Honors and awards:

- Fellow of the ACM
- Fellow of the IEEE

Institutional and professional service in last five years:

- Board Member, National Research Council's Army Research Laboratory Technical Assessment Board, 1996 -- Chair of Digitization Panel, 1999 --
- Organizer, NSF Workshop on Experimental Research in Computer Science, 1996
- Board Member, Computer Research Association, 1996 --
- Chair, CSTB Committee on Computer Literacy, National Research Council, 1997 - 1999
- Member, CISE Panel on Future Directions in Experimental Computer Science, 1998

States in which registered: none

Level of activity: professional societies - medium; research - high; consulting/summer work in industry - low.

Stepp, Marty

Lecturer (2006 Ass't)

Education:

- B.S., Computer Science, University of Arizona, 2001
- M.S., Computer Science, University of Arizona, 2003

Other Experience:

- 1 year as a developer at Microsoft Corporation

Principal publications of last five years:

- Building Java Programs, an introductory Java textbook (co-written with UW lecturer Stuart Reges)
- Computing Fundamentals with C#, an introductory C# textbook (co-written with Arizona lecturer Rick Mercer)

Scientific and professional societies of which a member:

- ACM SIGCSE

States in which registered: none

Level of activity: professional societies - low; research - low; consulting/summer work in industry - low.

Suciu, Dan

Associate professor (2000 Ass't, 2003 Assoc)

Education:

- M.S., Computer Science, Politechnic of Bucharest, Romania, 1982
- M.S., Mathematics, University of Bucharest, 1991
- Ph.D., , University of Pennsylvania, 1995

Other Experience:

- AT&T Labs, 1995-2000

Consulting, patents, etc.:

- US Patents:
- 6,134,553 Method for using region-sets to focus searches in hierarchical structures
- 6,076,087 Query evaluation on distributed semi-structured data
- 6,052,686 Database processing using schemas
- 5,999,926 View maintenance for unstructured databases
- 5,987,449 Queries on distributed unstructured databases
- 5,978,790 Method and apparatus for restructuring data in semi-structured databases
- 5,970,489 Method for using region-sets to focus searches in hierarchical structures
- 5,960,425 Database access system with optimizable expressions
- 5,956,720 Method and apparatus for web site management

Principal publications of last five years:

- Nilesh Dalvi, Dan Suciu, Efficient Query Evaluation on Probabilistic Databases. In VLDB, 2004
- Todd J. Green, Ashish Gupta, Gerome Miklau, Makoto Onizuka, Dan Suciu, Processing XML Streams with Deterministic Automata and Stream Indexes. Published in ACM TODS, vol. 29 , no. 4 , pp. 752-788 , December , 2004
- Nilesh Dalvi, Gerome Miklau, Dan Suciu, Asymptotic Conditional Probabilities for Conjunctive Queries. In ICDT, 2005
- Gerome Miklau, Dan Suciu, A Formal Analysis of Information Disclosure in Data Exchange. In SIGMOD, 2004
- Chris Re, Nilesh Dalvi, Dan Suciu, Efficient Top-k Query Evaluation on Probabilistic Data. in ICDE 2007
- Dan Suciu, Management of Probabilistic Data. in PODS 2007 (invited paper)
- Nilesh Dalvi, Dan Suciu, The Dichotomy of Conjunctive Queries on Probabilistic Structures. in PODS 2007

Scientific and professional societies of which a member:

- Member of ACM

Honors and awards:

- Best paper award at the ACM/SIGMOD International Conference on Management of Data, 2000
- Sloan Fellow, 2001

Institutional and professional service in last five years:

- co-PC Chair DBPL 2003
- PC Chair PODS 2004
- co-PC Chair XSym 2004
- co-PC Chair ICDT 2007
- Associate Editor for ACM Transactions on Database Systems (TODS)
- Associate Editor for ACM Transactions on Internet Technologies (TOIT)
- Associate Editor for Information Systems
- Program Committee member for SIGMOD, VLDB, DBPL

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Tanimoto, Steven

Professor (1977 Ass't, 1981 Assoc, 1987 Full)

Education:

- Ph.D., Electrical Engineering, Princeton University, 1975
- A.B., Visual and Environmental Studies, Harvard University, 1971
- M.A., Electrical Engineering, Princeton University, 1974
- M.S.E.E., Electrical Engineering, Princeton University, 1973

Other Experience:

- Assistant Professor, Univ. of Connecticut, 1975-1977

Consulting, patents, etc.:

- A Data Processing System Having a Pyramidal Array of Processors, U.S. Patent No. 4,622,632, issued on November 11, 1986 (Joseph J. Pfeiffer, Jr., co-inventor).

Principal publications of last five years:

- Tanimoto, S. L., 2007. Enhancing the prospects for educational data mining. Proc. Workshop on Educational Data Mining, Corfu, GR, June 2007.
- Tanimoto, S. L., Evans, N., and Carlson, A., 2007. Sequential input graphical model assessment diagrams for analysis of student activity data. Proc. ICALT 2007, the Int'l Conf. on Advanced Learning Technologies, Niigata, Japan, 2007.
- Tanimoto, S. L., Evans, N., and Carlson, A., 2007. Using graphical models to unobtrusively assess student performance in educational image processing activities. Proc. ICALT 2007, the Int'l Conf. on Advanced Learning Technologies, Niigata, Japan, 2007.
- Tanimoto, S. L., Hubbard, S., and Winn, W. 2005. Automatic textual feedback for guided inquiry learning. Proc. AIED 2005, Amsterdam, Netherlands, pp.662-669.
- Tanimoto, S. L. 2004. Transparent interfaces: Models and methods. Proc. Workshop on Invisible and Transparent Interfaces, Gallipoli, Italy, May.
- Tanimoto, S. L., Winn, W. D., and Teng, Chia-En. 2004. Using Event Logs to Study Conceptual Change in University Freshmen Studying Computer Science. Presentation at the American Educational Research Association Annual Meeting.
- Tanimoto, S. L. 2003. Programming in a Data Factory. Proc. Symposium on Visual Languages; International Symposia on Human-Centric Computing Languages and Environments, Auckland, NZ.
- Carlson, A., and Tanimoto, S. L. 2003. Learning to identify student preconceptions from text. In Burstein, J., and Leacock, C. (eds.), HLT-NAACL 2003 Workshop: Building Educational Applications Using Natural Language Processing, Edmonton, Alberta, Canada; Association for Computational Linguistics, pp.9-16.

Scientific and professional societies of which a member:

- IEEE Computer Society, Association for Computing Machinery

Honors and awards:

- IEEE Fellow. Fellow of the International Association for Pattern Recognition.

- IEEE Computer Society Outstanding Contribution Award.

Institutional and professional service in last five years:

- IEEE Computer Society Publications Board: Transactions Operations Committee Chair, 2007;
- Publications Board Member-at-large, 2006.
- International Symposium on Visual Languages and Human-Centric Computing, Steering Committee Chair, 2007;
- Program Committees: VL/HCC 2007, AVI 2008. International Conference on Pattern Recognition, 2008: Finance co-chair.
- Various review panels at the National Science Foundation.
- Associate Editor, Pattern Recognition, 1983-present.
- Advisory Editor, Journal of Visual Languages and Computing, 1990-present.

States in which registered: none

Level of activity: professional societies - high; research - high; consulting/summer work in industry - low.

Tompa, Martin

Professor (1978 Ass't, 1984 Assoc, 1989 Full)

Education:

- A.B., Applied Mathematics, Harvard University, 1974
- M.S., Computer Science, University of Toronto, 1975
- Ph.D., Computer Science, University of Toronto, 1978

Other Experience:

- 1985-87: Research Staff Member, Theory of Computation, IBM Research Division, Thomas J. Watson Research Center.
- 1987-89: Manager, Theory of Computation, IBM Research Division, Thomas J. Watson Research Center.

Consulting, patents, etc.:

- 1980: Consultant to Boeing Aerospace Company on VLSI design.
- 1993-94: Visiting Researcher, Microsoft Research.
- 1999: Consultant to Rosetta Inpharmatics.
- 2000-01: Visiting Researcher, ZymoGenetics.
- "Methods and apparatus for hinting a font for controlling stem width as font size and resolution of output device vary" (with D. Harel and E. Kohen). Microsoft Corporation. Patent No. 5,598,520, January 28, 1997.
- "Methods and system for controlling intercharacter spacing as font size and resolution of output device vary" (with D. Harel and E. Kohen). Microsoft Corporation. Patent No. 5,740,456, April 14, 1998.

Principal publications of last five years:

- G. Dantas, A. L. Watters, B. M. Lunde, Z. M. Eletr, N. G. Isern, T. Roseman, J. Lipfert, S. Doniach, M. Tompa, B. Kuhlman, B. L. Stoddard, G. Varani, and D. Baker, Mistranslation of a Computationally Designed Protein Yields an Exceptionally Stable Homodimer: Implications for Protein Engineering and Evolution. *Journal of Molecular Biology*, vol. 362, issue 5, 6 October 2006, 1004-1024.
- Nan Li and Martin Tompa, Analysis of Computational Approaches for Motif Discovery. *Algorithms for Molecular Biology*, vol. 1, no. 8, May 2006.
- Shane Neph and Martin Tompa, MicroFootPrinter: a Tool for Phylogenetic Footprinting in Prokaryotic Genomes. *Nucleic Acids Research*, vol. 34, July 2006, W366-W368.
- Amol Prakash and Martin Tompa, Discovery of Regulatory Elements in Vertebrates Through Comparative Genomics. *Nature Biotechnology*, vol. 23, no. 10, October 2005, 1249 - 1256.
- Amol Prakash and Martin Tompa, Statistics of Local Multiple Alignments. 13th Annual International Conference on Intelligent Systems for Molecular Biology, Detroit, MI, June 2005. *Bioinformatics*, vol. 21, June 2005, i344 - i350.
- M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. De Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu, Assessing Computational Tools for the Discovery of Transcription Factor Binding Sites. *Nature Biotechnology*, vol. 23, no. 1, January 2005, 137 - 144.
- Saurabh Sinha, Mathieu Blanchette, and Martin Tompa, PhyME: A Probabilistic Algorithm for Finding Motifs in Sets of Orthologous Sequences. *BMC Bioinformatics*, vol. 5, 2004, 170.
- Maria Shnyreva, William M. Weaver, Mathieu Blanchette, Scott L. Taylor, Martin Tompa, David R. Fitzpatrick, and Christopher B. Wilson, Evolutionarily Conserved Sequence Elements that Positively Regulate IFN- γ Expression in T Cells. *Proceedings of the National Academy of Science USA*, vol. 101, no.

34, August 2004, 12622-12627.

- Amol Prakash, Mathieu Blanchette, Saurabh Sinha, and Martin Tompa, Motif Discovery in Heterogeneous Sequence Data. Pacific Symposium on Biocomputing, Hawaii, January 2004, 348-359.
- Mathieu Blanchette and Martin Tompa, FootPrinter: a Program Designed for Phylogenetic Footprinting. Nucleic Acids Research, vol. 31, no. 13, July 2003, 3840-3842.
- Saurabh Sinha and Martin Tompa, YMF: a Program for Discovery of Novel Transcription Factor Binding Sites by Statistical Overrepresentation. Nucleic Acids Research, vol. 31, no. 13, July 2003, 3586-3588.
- Saurabh Sinha and Martin Tompa, Discovery of Novel Transcription Factor Binding Sites by Statistical Overrepresentation. Nucleic Acids Research, vol. 30, no. 24, December 2002, 5549-5560.
- Mathieu Blanchette and Martin Tompa, Discovery of Regulatory Elements by a Computational Method for Phylogenetic Footprinting. Genome Research, vol. 12, no. 5, May 2002, 739-748.
- Mathieu Blanchette, Benno Schwikowski, and Martin Tompa, Algorithms for Phylogenetic Footprinting. Journal of Computational Biology, vol. 9, no. 2, 2002, 211-223.
- Jeremy Buhler and Martin Tompa, Finding Motifs Using Random Projections. Journal of Computational Biology, vol. 9, no. 2, 2002, 225-242.
- Charles J. Colbourn, Alan C. H. Ling, and Martin Tompa, Construction of Optimal Quality Control for Oligo Arrays. Bioinformatics, vol. 18, no. 4, April 2002, 529-535.

Scientific and professional societies of which a member:

- International Society for Computational Biology

Honors and awards:

- 1984-86: Presidential Young Investigator Award
- 1998: ACM Undergraduate Teaching Award
- 1999: ACM Undergraduate Teaching Award

Institutional and professional service in last five years:

- Program Committee member, 2nd Workshop on Algorithms in Bioinformatics, Rome, Italy, September 2002.
- Program Committee member, Seventh Annual International Conference on Computational Biology, Berlin, Germany, April 2003.
- Program Committee member, Eighth Annual International Conference on Computational Biology, San Diego, CA, March 2004.
- Program Committee member, 13th Annual International Conference on Intelligent Systems for Molecular Biology, Detroit, MI, June 2005.
- Departmental service: Committee on restructuring the graduate examinations (2004-2006), Computational biology liaison (2001-present), Faculty lunch research organization (2004-2005).

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - high.

Wetherall, David

Assistant professor (1999 Ass't, 2004 Assoc)

Education:

- B.E., Electrical Engineering, University of Western Australia, 1989
- M.S., Computer Science, Massachusetts Institute of Technology, 1994
- E.E., Computer Science, Massachusetts Institute of Technology, 1995
- Ph.D., Computer Science, Massachusetts Institute of Technology, 1998

Other Experience:

- Director of Intel Research Seattle, 7/2006 to present

Principal publications of last five years:

- R. Mahajan, D. Wetherall, and T. Anderson, "Mutually Controlled Routing with Independent ISPs," 4th Symposium on Networked Systems Design and Implementation (NSDI 2007), Cambridge, MA, April 2007.
- R. Mahajan, M. Rodrig, D. Wetherall and J. Zahorjan, "Analyzing the MAC-level behavior of Wireless Networks in the Wild," ACM SIGCOMM 2006, Pisa, Italy, Sep. 2005. (Acceptance rate 12%)
- C. Reis, R. Mahajan, M. Rodrig, D. Wetherall and J. Zahorjan, "Measurement-based Models of Delivery and Interference in Static Wireless Networks," ACM SIGCOMM 2006, Pisa, Italy, Sep. 2005. (Acceptance rate 12%)
- X. Yang and D. Wetherall, "Source Selectable Path Diversity via Routing Deflections," ACM SIGCOMM 2006, Pisa, Italy, Sep. 2005. (Acceptance rate 12%)
- X. Yang, D. Wetherall and T. Anderson, "TVA: A DoS-limiting Network Architecture," ACM SIGCOMM 2005, Philadelphia, PA, Aug. 2005. (Acceptance rate 9%)
- R. Mahajan, D. Wetherall, and T. Anderson, "Negotiation-Based Routing Between Neighbor-ing ISPs," 2nd Symposium on Networked Systems Design and Implementation (NSDI 2005), Boston, MA, April 2005. (Acceptance rate 22%)
- R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Sustaining Cooperation in Multi-hop Wireless Networks," 2nd Symposium on Networked Systems Design and Implementation (NSDI 2005), Boston, MA, April 2005. (Acceptance rate 22%)
- K. Gummadi, H. Madhyastha, S. Gribble, H. Levy, and D. Wetherall, "Improving the Reliability of Internet Paths with One-hop Source Routing," 6th Symposium on Operating Systems Design and Implementation (OSDI 2004), San Francisco, CA, December 2004. (Acceptance rate 14%)
- R. Mahajan, N. Spring, D. Wetherall and T. Anderson, "User-level Internet Path Diagnosis," 19th ACM Symposium on Operating Systems Principles (SOSP'03), pp106-19, New York, Oct. 2003. (Acceptance rate 17%)
- P. Patel, A. Whitaker, D. Wetherall, J. Lepreau, and T. Stack "Upgrading Transport Protocols with Untrusted Mobile Code," 19th ACM Symposium on Operating Systems Principles (SOSP'03), pp1-14, New York, Oct. 2003. (Acceptance rate 17%)
- N. Spring, D. Wetherall and T. Anderson, "Scriptroute: A Public Internet Measurement Facility," 4th USENIX Symposium on Internet Technologies and Systems (USITS'03), pp225-38, Seattle, WA, Mar. 2003. (Acceptance rate 27%) Best Student Paper.
- S. Jain, R. Mahajan and D. Wetherall, "A Study of the Performance Potential of DHT-based Overlays," 4th USENIX Symposium on Internet Technologies and Systems (USITS'03), pp141-54, Seattle, WA, Mar. 2003. (Acceptance rate 27%)
- N. Spring, R. Mahajan, D. Wetherall and T. Anderson, "Measuring ISP Topologies with Rocketfuel," IEEE/ACM Transactions on Networking, 12(1):2-16, Feb. 2004. Selected for forwarding from ACM SIGCOMM'02. Bennett Prize Paper.
- R. Mahajan, D. Wetherall and T. Anderson, "Understanding BGP Misconfiguration," ACM SIGCOMM

2002, Pittsburgh, PA, Aug. 2002. (Acceptance rate 8%)

Scientific and professional societies of which a member:

- IEEE
- ACM
- Usenix

Honors and awards:

- IEEE Communications Society Bennett Prize, “Measuring ISP Topologies with Rocketfuel,” 2005.
- Sloan Fellow, 2004.
- 4th USENIX Symposium on Internet Technologies and Systems (USITS’03) Best Student Paper Award for “Scriptroute: A Public Internet Measurement Facility,” 2003.
- NSF CAREER Award, 2002.
- ACM SIGCOMM 2002 Best Student Paper Award (sole award out of 300 submissions) for “Measuring ISP Topologies with Rocketfuel,” 2002.
- Hackett Fellowship awarded for graduate study. 1991-94.

Institutional and professional service in last five years:

- Program Committee member, SIGCOMM 2007.
- Program Committee member, MobiSys 2007.
- Program Committee member, HotOS XI, 2007.
- Program Committee member, Passive and Active Measurement (PAM) 2007.
- Program Committee member, IEEE PerCom 2007.
- Program Committee member, SIGCOMM Workshop on Experimental Approaches to Wire-less Network Design and Analysis (E-WIND-05), 2005.
- Program Co-chair, Usenix/ACM NSDI 2005.
- Program Committee member, SIGCOMM Internet Measurement Conference 2004.
- Program Co-chair, ACM SIGCOMM 2003.
- Steering Committee member, HotNets Workshop, 2002-2004.
- Program Committee member, 5th USENIX OSDI, 2002.
- Co-founder and Program Co-chair, 1st Workshop on Hot Topics in Networks (HotNets), 2002.
- Program Committee member, OPENARCH, 2002.
- Program Committee member, ACM SIGCOMM 2002.

States in which registered: none

Level of activity: professional societies - low; research - high; consulting/summer work in industry - medium.

Whitaker, Andrew

Visiting Instructor

Education:

- B.S. Computer Science, Indiana University, 1999
- Ph.D., Computer Science, University of Washington, 2005

Other Experience:

- Postdoc, University of Washington CSE — January 2007 – Present
- Development Engineer, Amazon.com — October 2005 – November 2006
- Research Assistant, University of Washington Computer Science — June 2000 – September 2005
- Developer, Applied Research Labs; Austin, TX — summer 1998

Consulting, patents, etc.:

- None

Principal publications of last five years:

- Configuration Debugging as Search: Finding the Needle in the Haystack, Andrew Whitaker, Richard S. Cox, and Steven D. Gribble Proceedings of the 6th Symposium on Operating System Design and Implementation (OSDI) December 2004.
- Constructing Services With Interposable Virtual Hardware, Andrew Whitaker, Richard S. Cox, Marianne Shaw, and Steven D. Gribble Proceedings of the 1st Symposium on Network System Design and Implementation (NSDI) March 2004.
- Upgrading Transport Protocols Using Mobile Code' Parveen Patel, Andrew Whitaker, David Wetherall, Jay Lepreau, and Tim Stack, Proceedings of the 19th Symposium on Operating System Principles (SOSP) October 2003.
- Scale and Performance in the Denali Isolation Kernel, Andrew Whitaker, Marianne Shaw, Steven D. Gribble
- Proceedings of the 5th Symposium on Operating System Design and Implementation (OSDI) December 2002.
- Forwarding Without Loops in Icarus, Andrew Whitaker and David Wetherall, Proceedings of the 5th IEEE Conference on Open Architectures and Network Programming June 2002.

Scientific and professional societies of which a member:

Honors and awards:

Institutional and professional service in last five years:

States in which registered: None

Level of activity: professional societies - low; research - high; consulting/summer work in industry - low.

Weld, Daniel

Professor (1988 Ass't, 1993 Assoc, 1997 Full)

Education:

- Ph.D., Artificial Intelligence, Massachusetts Institute of Technology, 1988
- M.S., Computer Science, Massachusetts Institute of Technology, 1984
- B.S., Computer Science, Yale University, 1982
- B.A., Molecular Biophysics & Biochemistry, Yale University, 1982

Other Experience:

- Vice President of Engineering, Netbot Inc 1997
- Venture Partner, Madrona Venture Group 1999

Consulting, patents, etc.:

- Patents Pending:
- Method and System for Accessing an On-Line Store, Application 09/008,413 Doorenbos, Etzioni, Weld
- Method and System for Network Information Access, Application 60/035,844 Friedman, Weld, Kwok
- Method and System for Network Information Access, Application 9032-003 Selberg et al.
- Method and Apparatus for use in Accessing On-line Semi-structured Information, Application 08/982,857 Kushmerick, Doorenbos, Weld - filed 2 Dec 97

Principal publications of last five years:

- K. Gajos, J. Long and D. Weld, "Automatically Generating Custom User Interfaces for Users with Physical Disabilities" Eighth International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS-06), Portland, OR, October, 2006.
- K. Gajos, M. Czerwinski, D. Tan, and D. Weld, "Exploring the Design Space for Adaptive Graphical User Interfaces" Advanced Visual Interfaces (AVI-06), Venice, Italy, May, 2006.
- K. Gajos and D. Weld, "Preference Elicitation for Interface Optimization" ACM User Interface Software and Technology (UIST-05), Seattle, WA, October 2005.
- Krzysztof Gajos, David Christianson, Raphael Hoffmann, Tal Shaked, Kiera Henning, Jing Jing Long, and Daniel S. Weld. "Fast And Robust Interface Generation for Ubiquitous Applications." Seventh International Conference on Ubiquitous Computing (UBICOMP'05). Tokyo, Japan, September, 2005.
- Corin R. Anderson, Pedro Domingos, and Daniel S. Weld, "Relational Markov Models and their Application to Adaptive Web Navigation." Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). 2002.
- Mausam and Weld, D., "Probabilistic Temporal Planning with Uncertain Durations," National Conference on Artificial Intelligence (AAAI-06). Boston, MA, 2006.
- Mausam and Weld, D., "Challenges for Temporal Planning with Uncertain Durations," International Conference on Automated Planning and Scheduling (ICAPS-06). Cumbria, UK. June 2006.
- K. Gopalratnam and H. Kautz and D. Weld "Extending Continuous Time Bayesian Networks", AAAI-05, Pittsburgh, PA, USA. August 2005.
- Mausam and Daniel Weld "Concurrent Probabilistic Temporal Planning" International Conference on Automated Planning and Scheduling (ICAPS). Monterey, CA, USA. June 2005.
- Mausam and D. Weld, "Solving Concurrent Markov Decision Processes," AAAI 2004, San Jose, CA July 2004.
- Mausam and D. Weld, "Concurrent Probabilistic Temporal Planning: Initial Results," AAAI 2004 workshop on Learning and Planning in Markov Processes - Advances and Challenges, San Jose, CA July 2004.

- Downey, D., Etzioni, O., Soderland, S. and Weld, D., "Learning Text Patterns for Web Information Extraction and Assessment," AAAI 2004 workshop on Adaptive Text Extraction and Mining, San Jose, CA July 2004.
- Ives, Z., Halevy, A. and Weld, D., "Adapting to Source Properties in Processing Data-Integration Queries," 2004 ACM SIGMOD Conference, Paris, France, June 2004.
- Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A-M., Shaked, T., Soderland, S., Weld, D. and Yates, A., "Web-scale Information Extraction in KnowItAll," WWW 2004, New York, NY, May 2004.
- Z. Ives, A. Halevy, and D. Weld "An XML Query Engine for Network-Bound Data" VLDB Journal, Vol. 11 No. 4, 2002.

Scientific and professional societies of which a member:

- American Association for Artificial Intelligence
- Association for Computing Machinery
- Institute of Electrical and Electronics Engineers

Honors and awards:

- Fellow, Association of Computing Machinery, 2006
- WRT / TJ Cable Endowed Professorship, 1999
- Fellow, American Association of Artificial Intelligence, 1999
- Edge Award for best use of intelligent technology (in Jango), WebINNOVATION Show, 1997
- New Innovator's Award (for Jango), CommerceNet, 1997
- Award for Technological Innovation in Computer Software (Internet Softbot one of five finalists), Discover Magazine, 1995
- Young Investigator Award, Office of Naval Research, 1990
- Presidential Young Investigator Award, National Science Foundation, 1989

Institutional and professional service in last five years:

- Elected Councillor of the American Association of Artificial Intelligence (AAAI) (1994--1997)
- Member of Advisory Board, Journal of Artificial Intelligence Research (1992--)
- Member and Editor, AAAI/NSF Committee on Intelligence in the NII (1994)
- Member, AAAI/ARPA Committee on Twenty-First Century Intelligent Systems (1994)
- ICAPS Workshop Chair (2007)
- Consultant & Venture Partner, Madrona Venture Group (2001-)
- Founder, Asta Networks (200)
- Founder, Nimble.com (1999)
- Founder, AdRelevance Inc. (1998)
- Consultant, Excite Inc. (1998)
- Founder, Netbot Inc. (1996) Creator of Jango shopping search

States in which registered: none

Level of activity: professional societies - high; research - high; consulting/summer work in industry - high.

Zahorjan, John

Professor (1980 Ass't, 1985 Assoc, 1989 Full)

Education:

- Ph.D., Computer Science, University of Toronto, 1980
- M.Sc., Computer Science, University of Toronto, 1976
- Sc.B., Applied Mathematics, Brown University, 1975

Principal publications of last five years:

- Ratul Mahajan, Maya Rodrid, David Wetherall, and John Zahorjan, "Analyzing the MAC-level Behavior of Wireless Networks in the Wild", Proceedings of ACM SIGCOMM, September 2006.
- Charlie Reiss, Ratul Mahajan, Maya Rodrid, David Wetherall, and John Zahorjan, "Measurement-Based Models of Delivery and Interference in Static Wireless Networks", Proceedings of ACM SIGCOMM, September 2006.
- Thomas Anderson, Andrew Collins, Arvind Krishnamurthy, and John Zahorjan, "PCP: Efficient Endpoint Congestion Control," 3rd Symposium on Networked Systems Design & Implementation (NSDI), May 2006.
- Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan, "Sustaining Cooperation in Wireless Networks." 2nd Symposium on Networked Systems Design & Implementation (NSDI), May 2005.
- Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19), October 2003.

Institutional and professional service in last five years:

- Technical Program Committee Member, WiNMee/WiTMeMo 2007 (3rd Intl. Workshop on Wireless Network Measurement / 3rd Intl. Workshop on Wireless Traffic Measurements and Modeling).
- Technical Program Committee Member, PerCom 2006 (4th Annual IEEE International Conference on Pervasive Computing and Communications).
- Technical Program Committee Member, PerCom 2005 (3rd Annual IEEE International Conference on Pervasive Computing and Communications).

States in which registered: none

Level of activity: professional societies - low; research - medium; consulting/summer work in industry - low.

D. Supplementary Material

1. Current Computer Engineering program requirements.
2. New Computer Engineering program requirements, effective Fall 2007.
3. Revised Table I-1a – Basic-Level Curriculum for Computer Engineering
4. Instructions for our targeted outcome assessment process.

Computer Engineering program requirements in effect 2004-2007

Computer Engineering Graduation Requirements

University of Washington

The graduation requirements shown below are subject to change.

For more information, see the Undergraduate Handbook, available online at
http://www.cs.washington.edu/education/ugrad/current/degree_requirements.html

General Education Component

Written & Oral Communication (12 credits)

- *English Composition (5)
- TC 231 Intro. to Technical Writing (3)
- TC 333 Adv. Tech. Writing & Oral Pres. (4)

Areas of Knowledge (30 credits)

- Visual, Literary, and Performing Arts (10-20)
- Individuals and Societies (10-20)

Mathematics & Science Component

Mathematics (19-22 credits)

- *Math 124, 125, 126 or 134, 135, 136 (honors) (15)
Calculus with Analytical Geometry
- Math 308 or 318 (waived if 136 taken) (3)
- Stat 390, 391, or 394&395 (4)
-extra 2 credits from 394/395 count as
CSE senior elective credits

Natural Sciences (20 credits)

- *Phys 121 Mechanics (5)
- *Phys 122 Electromagnetism &
Oscillatory Motion (5)
- 10 additional credits from the list (10)
of approved natural science courses in the
CS&E Handbook

* Denotes prerequisites (must be fully completed before application date). Regardless of AP credit, at least one calculus or post-calculus math course and one approved natural science course must be completed prior to applying to the department.

Computer Engineering Component

Required (47 credits)

- *CSE 142 Computer Programming I (4)
- *CSE 143 Computer Programming II (5)
- CSE 303 Concepts & Tools for Soft. Dev. (3)
- CSE 321 Discrete Structures (4)
- CSE 322 Intro to Formal Models (3)
- CSE 326 Data Structures (4)
- CSE 341 Programming Languages (4)
- CSE 370 Intro to Digital Design (4)
- CSE 378 Machine Org & Assembly Lang. (4)
- EE 215 Intro to Electrical Engineering (4)
- CSE 451 Operating Systems (4)
- CSE 461 Intro to Networks (4)

Students must complete either the hardware or the software specialization.

Hardware Specialization (24 credits)

- EE 233 Circuit Theory (5)
- CSE 466 Software for Embedded Systems (4)
- CSE 467 Advanced Digital Design (4)
- CSE 477 Hardware Design Capstone (5)
- Select at least 6 credits from courses on
the approved senior elective course list in the
CS&E Handbook (6)

Software Specialization (22-24 credits)

- CSE 403 Software Engineering (4)
- Two from CSE 401, 421, 444, 466, or 471 (6-8)
- Software Design Capstone (5)
- Select at least 7 credits from courses on the
approved senior elective course list in the CS&E
Handbook (7)

**Free Electives to bring total credits up to the 180
required for graduation (25-32 credits)**

The minimum acceptable grade for any required or elective CSE course is 2.0. A student's overall GPA must not fall below a 2.0.

New Computer Engineering program requirements, effective Fall 2007.

Computer Engineering Graduation Requirements

University of Washington

The graduation requirements shown below are subject to change.

For more information, see the Undergraduate Handbook, available online at
http://www.cs.washington.edu/education/ugrad/current/degree_requirements.html

General Education Component

Written & Oral Communication (12 credits)

- ☐ *English Composition (5)
- ☐ TC 231 Intro. to Technical Writing (3)
- ☐ TC 333 Adv. Tech. Writing & Oral Pres. (4)

Areas of Knowledge (30 credits)

- ☐ Visual, Literary, and Performing Arts (10-20)
- ☐ Individuals and Societies (10-20)

Mathematics & Science Component

Mathematics (19-22 credits)

- ☐ *Math 124, 125, 126 or 134, 135, 136 (honors) (15)
Calculus with Analytical Geometry
- ☐ Math 308 or 318 (waived if 136 taken) (3)
- ☐ Stat 390, 391, or 394&395 (4)
-extra 2 credits from 394/395 count as
free elective credits

Natural Sciences (20 credits)

- ☐ *Phys 121 Mechanics (5)
- ☐ *Phys 122 Electromagnetism &
Oscillatory Motion (5)
- ☐ 10 additional credits from the list
of approved natural science courses in the
CS&E Handbook (10)

*** Denotes prerequisites (must be fully completed before application date). Regardless of AP credit, at least one calculus or post-calculus math course and one approved natural science course must be completed prior to applying to the department.**

Computer Engineering Component

Required (47 credits)

- ☐ *CSE 142 Computer Programming I (4)
- ☐ *CSE 143 Computer Programming II (5)
- ☐ CSE 303 Concepts & Tools for Soft. Dev. (3)
- ☐ CSE 321 Discrete Structures (4)
- ☐ CSE 322 Intro to Formal Models (3)
- ☐ CSE 326 Data Structures (4)
- ☐ CSE 341 Programming Languages (4)
- ☐ CSE 370 Intro to Digital Design (4)
- ☐ CSE 378 Machine Org & Assembly Lang. (4)
- ☐ EE 215 Intro to Electrical Engineering (4)
- ☐ CSE 451 Operating Systems (4)
- ☐ CSE 461 Intro to Networks (4)

Students must complete either the hardware or the software specialization.

Hardware Specialization (28 credits)

- ☐ EE 233 Circuit Theory (5)
- ☐ CSE 466 Software for Embedded Systems (4)
- ☐ CSE 467 Advanced Digital Design (4)
- ☐ CSE 471 Computer Design and Organization (4)
- ☐ CSE 477 Hardware Design Capstone (5)
- ☐ Select at least 6 credits from courses on
the approved Computer Engineering senior elective
course list in the CS&E Handbook (6)

Software Specialization (27 credits)

- ☐ CSE 403 Software Engineering (4)
- ☐ Three from CSE 401, 421, 444, 466, 471, 484 (9-12)
- ☐ Software Design Capstone (5)
- ☐ Select additional courses from the approved
Computer Engineering senior elective course list in the
CS&E Handbook to bring the total for the Software
Specialization to 27 credits. (6-9)

Free Electives to bring total credits up to the 180 required for graduation (21-25 credits)

The minimum grade for required math, science, written or oral communication, CE senior electives, and computer engineering coursework is a 2.0. A student's cumulative GPA must not fall below a 2.0.

Table I-1a. Basic-Level Curriculum – Computer Engineering
(New requirements to take effect Fall 2007)

Year; Semester or Quarter	Course (Department, Number, Title)	Category (Credit Hours)			
		Math & Basic Sciences	Engineering Topics Check if Contains <i>Significant Design</i> (✓)	General Education	Other
1 Yr/ 1 st Qtr	Math 124 Calc w/Anal Geometry	5	()		
	Natural Science Elective	5	()		
	English Composition		()	5	
1 Yr/ 2 nd Qtr			()		
	Math 125 Calc w/Anal Geometry	5	()		
	Phys 121/131 Mechanics	5	()		
	VLPA/I&S Elective		()	5	
1 Yr/ 3 rd Qtr			()		
	Math 126 Calc w/Anal Geometry	5	()		
	Physics 122/132 Electromagnetism	5	()		
	VLPA/I&S Elective		()	5	
2 Yr/ 1 st Qtr	Natural Science Elective	5	()		
	CSE 142 Computer Programming I		4 (✓)		
	VLPA/I&S Elective		()	5	
			()		
2 Yr/ 2 nd Qtr	Math 308 Linear Algebra	3	()		
	CSE 143 Computer Programming II		5 (✓)		
	VLPA/I&S Elective		()	4	
	VLPA/I&S Elective		()	4	
2 Yr/ 3 rd Qtr			()		
	Math/Stat 390 Prob & Statistics	4	()		
	EE 215 Fundamentals of EE		4 (✓)		
	TC 231 Intro. to Technical Writing		()	3	
	VLPA/I&S Elective		()	4	
3 Yr/ 1 st Qtr	CSE 303 Concepts&Tools for SW		3 ()		
	CSE 370 Intro to Digital Design		4 (✓)		
	EE 233 Circuit Theory (HW option)		5 (✓)		
	Free Elective		()		3-8
			()		
3 Yr/ 2 nd Qtr	CSE 378 Mach Org/Assembly Lang		4 (✓)		
	CSE 321 Discrete Structures	2	2 ()		
	VLPA/I&S Elective		()	5	
			()		
3 Yr/ 3 rd Qtr	CSE 341 Programming Languages		4 (✓)		
	CSE 322 Formal Models	1	2 ()		
	CSE 326 Data Structures		4 (✓)		
	Free Elective		()		4

Table I-1a. Basic-Level Curriculum – Computer Engineering (continued)
(New requirements to take effect Fall 2007)

Year; Semester or Quarter	Course (Department, Number, Title)	Category (Credit Hours)			
		Math & Basic Science	Engineering Topics <i>Check if Contains Significant Design</i> (✓)	General Education	Other
4 Yr/ 1 st Qtr Hardware Option	CSE 451 Operating Systems		4 (✓)		
	CSE 461 Networks		4 (✓)		
	CSE 467 Advanced Digital Design		4 (✓)		
	Free Elective		()		3
			()		
4 Yr/ 2 nd Qtr	CSE 466 SW for Embedded Systems		4 (✓)		
	CSE 471 Computer Design		4 (✓)		
	CompE Elective		3 ()		
	Free Elective		()		4
			()		
4 Yr/ 3 rd Qtr	CSE 477 Digital Systems Capstone		5 (✓)		
	TC 333 Adv. Tech Writing & Oral Presentation		()	4	
	CompE Elective		3 ()		
	Free Elective		()		3
		[45]	[72]	[44]	
4 Yr/ 1 st Qtr Software Option	CSE 403 Software Engineering		4 (✓)		
	CSE 451 Operating Systems		4 (✓)		
	CompE Elective		3-4 ()		
	One of CSE 401 (Compilers), 421 (Algorithms), 444 (Databases), 464 (Security), 466 (Embedded Systems), 471 (Computer Design)		3-4 (✓)		
4 Yr/ 2 nd Qtr	CSE 461 Networks		4 (✓)		
	Two of CSE 401, 421, 444, 464, 466, 471		6-8 (✓)		
	Free Elective		()		3-5
			()		
4 Yr/ 3 rd Qtr	CSE 4xx Software Systems Capstone		5 (✓)		
	TC 333 Adv. Tech Writing & Oral Presentation		()	4	
	One to Two CompE Electives		3-6 ()		
	Free Elective				0-3
TOTALS-ABET BASIC-LEVEL REQUIREMENTS		45	71-72	42	
OVERALL TOTAL FOR DEGREE					
PERCENT OF TOTAL					
Totals must	Minimum quarter credit hours	45 hrs	67.5 hrs		
satisfy one set	Minimum percentage	25%	37.5 %		

Assessing ABET Outcomes in Classes

The ABET outcomes are a set of goals that we want our students to achieve by the time they graduate from our program. We assess to what extent our students meet these goals so that we can make changes to improve our program. We use a variety of methods to assess outcomes including feedback from our students and alumni and feedback from our friends from industry where students do coops and internships. However, the most meaningful assessment is a direct measurement of student performance in classes for specific outcomes.

The ABET Outcomes

Students should demonstrate:

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice.
- (l) knowledge of probability and statistics
- (m) knowledge of discrete mathematics

Assessing Outcomes in Courses

Although many outcomes are covered in many courses, we will limit assessment of outcomes to only a few courses that emphasize those outcomes. In so doing, we assess most outcomes in just three courses. This means that each course assesses at most three outcomes with the exception of the capstones, which assess almost all.

The table on the following page lists the outcomes to be assessed in each course. To some extent the choice of outcomes for each course can be adjusted if appropriate: this is a reasonable starting point. The outcomes f, j, i, and j are more difficult to find a home than the others because they fall outside our traditional course material. We will have to use some creativity in assessing these outcomes.

Exactly how the assessment is done will largely up to individual instructors, but course coordinators should coordinate this for the core courses. I'll suggest a framework for doing assessment, along with some examples. With some planning, the amount of effort spent on assessment should be minimal.

Table I : The outcomes that should be assessed in each course. Please let me know if you think that this list should be adjusted.

Course	Outcomes Assessed	Additional Outcomes Covered
CSE 303 – Concepts and Tools for SW Dev.	f, h, j	g, k
CSE 321 – Discrete Structures	l, m	a
CSE 322 – Introduction to Formal Models	m	a, c, e
CSE 326 – Data Structures	m	a, e, k
CSE 341 – Programming Languages	k	e
CSE 370 – Introduction to Digital Design	a	c, e, k
CSE 378 – Machine Org. and Assembly Lang.	b, c	a, e, k
CSE 451 – Operating Systems	b, d, e	a, c, k
CSE 461 – Networking	c, d, e	a, b, k
HW Track Courses		
CSE 466 – Embedded Systems	a, c, k	b, e
CSE 467 – Advanced Digital Design	a, c, k	b, e
SW Track Courses		
CSE 403 – Software Engineering	a, k	b, c, d, e, j
CSE 401 – Introduction to Compilers	a, k	c, e
CSE 421 – Introduction to Algorithms	a, e	m
CSE 444 – Databases	a, k	c, e, j
CSE 471 – Computer Design	a, k	b, c, e
Capstones	b, c, d, e, f, g, h, i, j	a, k

Assessment Strategies

The recommended assessment technique is to categorize student work into four categories:

novice (has not yet attained the outcome)

developing (has attained the outcome, but needs more practice)

competent (meets our expectations)

exemplary (exceeds our expectations)

How this is done will vary from outcome to outcome, but keeping in mind that assessment should result in a simple histogram of the class by these four categories will simplify things.

Outcomes a, b, c, e, k, l, and m

Assessing the more technically-oriented outcomes is best done using targeted assignment and test questions. In most cases it will be straightforward to map an outcome onto

be assessed or just individual questions. This does not need to be precise as long as it captures the overall achievement of the students for that outcome. Here is the suggested procedure:

Identify assignments, tests or specific questions that relate to the outcome. This will require some planning ahead.

After an assignment or test is graded, go back through it and assign each to one of the four categories to generate a histogram for the class. Categorizing student work could be a subjective evaluation of the instructor/TA, or done based on the grade assigned to the question/assignment/test, e.g. 0-50=novice, 50-70=developing, 70-90=competent, 90-100=exemplary.

Make copies of student work that represents students at the low, middle and high end of the class, excluding outliers at either end.

Submit a copy of the assignment/test, the sample copies, and the assessment histogram.

Outcome d (multi-disciplinary teams) [Capstones]

The following rubric can be used to assess students' ability to work on multi-disciplinary teams. Note that "multi-disciplinary" can be interpreted rather broadly to mean different tasks or skills for each team member. Again, students should be assessed using the four categories. Here are some criteria that can be used to make this assessment:

- 1) Team project management: setting and following collective goals and ground rules; setting timelines to meet critical path requirements; attendance and participation in team meetings.
- 2) Team productivity: responsible and appropriate delegation of work; individual contributions of effort and time; information exchange between members; collaboration in analysis of information and production of deliverables.
- 3) Team cohesiveness: understand, include and respect ideas/perspectives of all members; deal successfully with personality problems and conflicts; team report reflects integration of different members' content into a coherent voice.

Outcome f (ethics) [303, Capstones]

One method for assessing the students understanding of ethical responsibility is through an essay. For example, students can be asked to identify the ethical issues that arise in a specific situation and then to discuss the different possible courses of action and their ethical consequences. Scoring is necessarily objective, but can be done using the four categories outlined above.

Outcome g (communication) [Capstones]

Student writing is assessed every year or two by a committee that evaluates samples of student writing across several courses. Capstone courses assess both written and oral communication. The following criteria can be used to judge the student's ability to communicate:

Mechanics

- Mastery of standard English grammar, punctuation, and spelling

Organization

- Ability to organize and present material in an effective order
- Ability to partition the text into appropriate paragraphs and sections by topic

Content

- Ability to focus on most relevant information
- Ability to explain difficult concepts with good use of examples and figures
- Includes all pertinent material

Document Design

- Demonstrates cohesion by graphic means (headings, white space) in documents
- Uses parallel structure with heading and subheadings
- Includes basic graphics

Citation

- Ability to identify and cite relevant sources
- Ability to follow standard citation rules

Oral Presentation

- Ability to organize material into a coherent story
- Good use of slides for text and figures
- Good poise and use of voice
- Adjusts to level of audience
- Ability to answer questions well

Outcome h (impact of computer engineering solutions) [303, Capstones]

Our students should be able to discuss intelligently the impact of specific computer engineering solutions on society and the environment. An example of assessing this outcome is to ask students to describe the known or potential impacts of a specific design and assess their responses in terms of relevance, detail and analysis.

Outcome i (life-long learning) [Capstones]

This outcome attempts to assess whether students understand the importance of continuous learning throughout life and have the skills to do it. Skills that we can assess in the capstone courses include the ability to find, learn and apply new information and skills to solve new problems. This can be done, for example, by including a section in the project report where students discuss the research they had to conduct, how they located the information they needed, and how it was applied in the project.

Outcome j (knowledge of contemporary issues) [303, Capstones]

This outcome can be assessed directly via essays similar to outcome (f) and by asking students to include in their reports a discussion of the contemporary issues related to their project.