

**Program Self-Study Report  
for the  
Bachelor of Science  
in  
Computer Engineering**

**University of Washington**

**4 June 2001**

# Table of Contents

<b>Program Self-Study Report</b> .....	
A. Background Information .....	1
1. Degree Titles .....	1
2. Program Modes .....	1
3. Actions to Correct Previous Deficiencies .....	1
4. Department Culture and Administrative Structure .....	2
5. Department Constituencies and Feedback Loops .....	4
6. Examples of the Feedback Loops Working .....	5
B. Accreditation Summary.....	7
1. Students.....	9
2. Program Educational Objectives.....	12
3. Program Outcomes and Assessment.....	16
4. Professional Component .....	31
5. Faculty.....	32
6. Facilities .....	33
7. Institutional Support and Financial Resources.....	35
8. Program Criteria.....	36
9. Cooperative Education Criteria.....	37
10. General Advanced-Level Program .....	38
<b>Appendix I - Additional Program Information</b> .....	
A. Tabular Data for Program.....	39
Table A.1. Basic Curriculum.....	40
Table A.2. Course and Section Size Summary .....	42
Table A.3. Faculty Workload Summary.....	45
Table A.4. Faculty Analysis .....	48
Table A.5. Support Expenditures .....	52
B. Course Syllabi .....	53
C. Faculty Resumes.....	97
D. Student exit survey comments for the past two years .....	196

# Brotman Award for Instructional Excellence

(University of Washington Departmental Education Award)

## Reflective Statement Prepared for CS&E Nomination

*CS&E won the Inaugural Award in Spring 1999*

CSE's educational efforts are shaped by a four-point philosophy:

- We believe that universities are, first and foremost, educational institutions, and that faculty members are, first and foremost, educators. Students are the "product" of the university and its faculty, in both senses of the word: they are the "output," but more importantly they are the "multiplier" that provides leverage for every faculty member. There are other ways to obtain leverage – one can, for example, conduct research without student involvement. But if you seek your primary leverage in some way other than by educating students, then you will be most effective in some other environment.
- We believe that a research university is a unique institution that can provide a unique undergraduate education – an education in which bright and committed students are brought to the very forefront of knowledge, closely mentored by faculty who are (with their students) working to redefine that forefront. More importantly, we believe that a research university must provide this kind of education, because no other institution can, and because any other kind of education can be provided better and/or cheaper elsewhere. We believe in differentiation: there is something that only the University of Washington and its peers can do, and we must focus on it. "If we're not striving to seamlessly integrate research and education, we're screwing up."
- We believe in taking a holistic view. Today's K-12 students are tomorrow's UW students; we must give them the tools to succeed. The citizens of Washington and their elected representatives pay the bills; they are entitled to a clear explanation of what we are trying to accomplish. The high-technology companies in our region, and our colleagues elsewhere at the University of Washington, represent enormous competitive advantages for us (and we for them); we must reach out in many ways. The future of the University of Washington is inextricably linked to the future of our region; creating a climate conducive to a 21st-century technology-based economy serves everyone's interests. Finally, special responsibilities fall to computer science programs as we approach the millenium, because of the role that computer science is playing in transforming all aspects of our lives; we must rise to these responsibilities.
- Finally, we believe that the University of Washington's highly capable students, staff, and faculty will respond to encouragement and example, striving for excellence in response to high expectations.

Our educational efforts and approaches follow directly from these principles. We strive to create the best possible educational experience for our students – one that benefits from, and that benefits, our position as a top-ten research program. We invest in our introductory courses because knowledge of computing is fundamental to success in the modern world, and because these courses are the "attraction waters" for our major. We aggressively recruit, advise, tutor, and mentor students, because we want a diverse collection of the University of Washington's finest students in our program, and we want these students to succeed. We encourage our undergraduates to work alongside faculty and graduate students as TAs, because this benefits both the students taking the courses and the students who TA them, and creates a "learning community" that extends from the youngest student to the oldest faculty member. We similarly encourage our undergraduates to work alongside faculty and graduate students as RAs, because this is one of many ways in which these students benefit from the unique type of education that only a research university can provide. We facilitate co-op and internship employment because, if properly integrated, it teaches the students things that are complementary to those they learn in our program. We constantly introduce new "Capstone Design

Courses" (many of which are interdisciplinary) because our field is advancing at a remarkable pace, and because these courses provide an unparalleled opportunity for students to synthesize what they have learned throughout their studies. We employ a wide range of "carrots" to encourage outstanding teaching, because encouragement and example work best: a departmental TA award, a departmental faculty teaching award, nomination of faculty and students for University and national recognition, quarterly student evaluations and annual peer evaluations for all faculty, quarterly circulation of a histogram of student evaluations for faculty and for TAs, and more. We invest aggressively in educational technology because it allows us to reach a broad audience of students and citizens, and because we believe that ultimately it will change the nature of education, allowing faculty members to spend more of their time doing the things that only they can do.

That, in fact, is our overriding objective: to do the things that only we can do, and to do them as well as they can be done.

Edward D. Lazowska, Chair  
Spring 1999

## 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 options, namely, hardware and embedded software, which were instituted beginning in the fall of 1999.

The Department also offers a **Bachelor of Science in Computer Science** degree through the College of Arts & Sciences.

### 2. Program Modes

Both of the department's degree programs require students to be enrolled **full-time on-campus**. Courses are offered during **day-time** hours during 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 special medical leaves.

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

### 3. Actions to Correct Previous Shortcomings

The previous ABET review (in 1996) found a "very strong program" with only one minor issue "with respect to the requirements of the ABET engineering criteria and applicable program criteria". The issue was the degree of exposure our students have to ethical, social, economic, and safety considerations. Although our curriculum does include general course categories relevant to these topics, the reviewers found that it was possible for students to graduate without guaranteed exposure to these topics.

Since 1996, we have been working to remedy this situation. We have experimented with several approaches. On an experimental basis, we have introduced course modules on ethics and societal impact of information technology into several of our senior-level courses as well as our large introductory programming courses. We have prototyped (for two quarters) a weekly seminar with guest speakers from around the campus and beyond. Finally, we have identified several courses offered at the University that address these issues.

Our Curriculum Committee's recommendation is to insert appropriate modules in the majority of our program's courses. Our faculty feels strongly that they can't "teach" ethics but can bring up issues as they arise during the progression of a course. Some examples are provided in the table below for many of the program's required and elective courses.

Course	Appropriate Ethics/Societal Impact Topic
software engineering	safety, reliability, and risk
embedded software	safety; user control; ubiquitous devices/sensors; privacy
databases	privacy
networks	domain name controversies, digital divide
graphics	it used to be that pictures didn't lie
programming languages	language/security interactions
algorithms	encryption - controversies & policy questions
operating systems	security; hacking; anti-trust
capstone design	professional ethics; IP issues; surveillance and privacy (embedded systems); sex and violence in video games and its effects (software design); safety/reliability (hardware design)

We have also expanded our list of recommended humanities/social-sciences (VLPA) electives to include many courses that treat these issues directly and are offered by a variety of departments on campus. We plan to continue a one quarter per year seminar with invited speakers from around campus and the local computing industry. Finally, we plan on adding a module to our capstone design courses and/or the senior-level technical communications course (in which students often enroll contemporaneously and we are working to link more tightly). Students will be asked to write a paper or a portion of their final report on the societal impact of their capstone projects (which can be in areas as diverse as chip design, hardware design, embedded system design, and distributed software design).

We feel strongly that it is important to expose students to ethical issues throughout the curriculum and not in separate, add-on courses that only serve to reinforce an often-perceived separation between technology and societal issues.

At the institutional (College) level, the primary criticism in the previous ABET review was that “the institution still must describe how the design experience is developed and integrated throughout each curriculum, show that it is consistent with the objectives of each program, and identify the major, meaningful design experiences in the curriculum of each engineering program”. In the remainder of this self-study it will be evident that these issues are not now (nor were they ever) a concern for this particular program.

#### 4. Department Culture and Administrative Structure

Collaboration and consensus are the strengths of our departmental culture. The departmental is not hierarchically organized. Rather, committees appointed by the chair, in consultation with an elected Executive Committee, present findings and recommendations to the entire faculty after data gathering and development of alternative proposals and solutions. The committee most relevant to this self-study is the Curriculum Committee that includes faculty, advising staff, undergraduates, and graduate students. The External Advisory Committee, consisting of leading academics and industry researchers, meets yearly to advise the department on its research and educational missions.

The department has two Associate Chairs whose duties are commensurate with the department’s strongest needs. This not only allows department resources to be brought to bear where they can do the most good but also tailors the duties of the Associate Chairs to the particular strengths and interests of the faculty members who take on the position. Currently these positions are dedicated primarily to Educational Outreach and Educational Technology and Distance Learning.

Undergraduate and graduate students have frequent interactions with the faculty through departmental social events (both sponsored by the department and student organizations). The Chair meets with undergraduates once a month at a pizza lunch that is attended by approximately 50 students each time. Although the time at this session is limited to approximately one hour, we have focused discussion topics (conversion of introductory programming from C++ to Java was a recent topic) and many students continue to communicate afterwards via e-mail to specific faculty (leading the discussion), the chair (who is almost always present), and/or advising staff. Graduate students also have frequent and similar interactions.

Administrative staff is supervised primarily by the clusters of faculty they each support. The main office, financial, and facilities staff are centrally supervised. The Lab Director supervises technical staff members who are organized into three separate groups: infrastructure development, operations, and educational support.

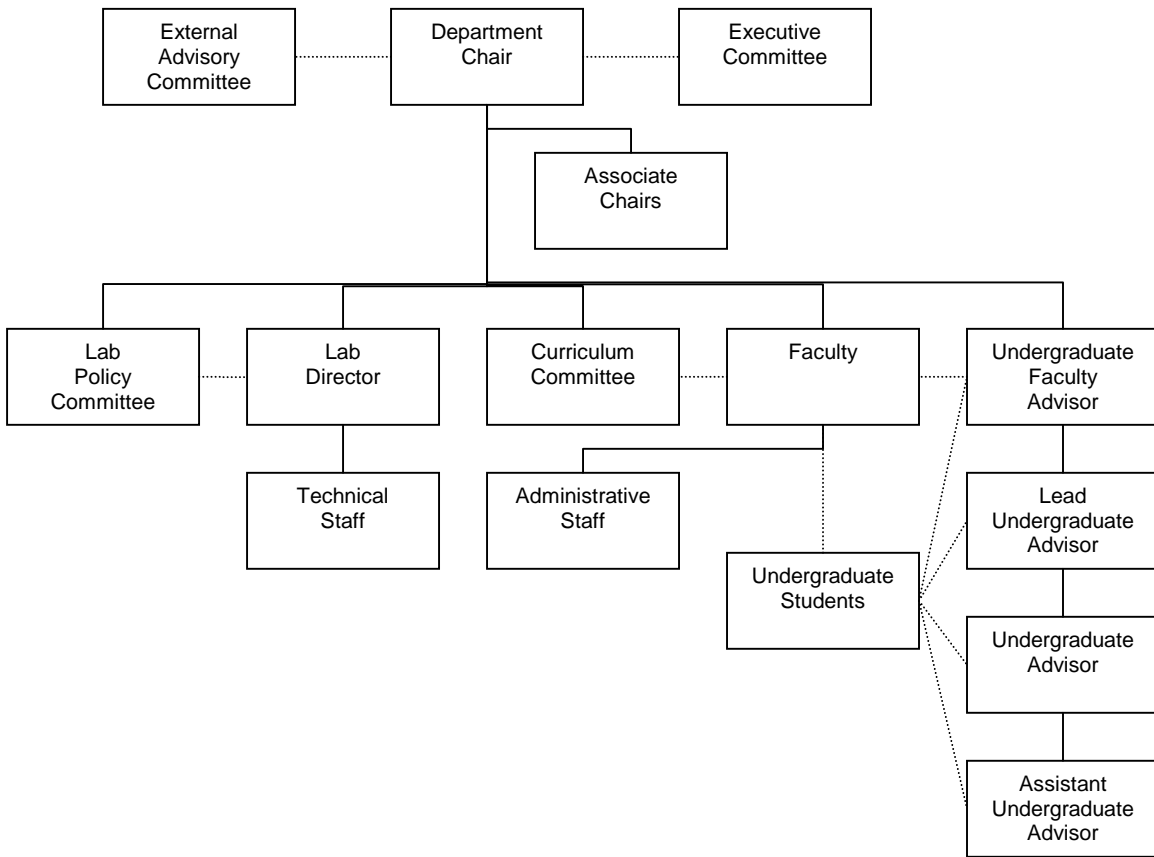


Figure 1. Department administrative structure related to the undergraduate program. Solid lines describe the reporting structure. Dashed lines describe consultation.

The Undergraduate Advising Unit, as shown in the chart above, is the responsibility of the Undergraduate Faculty Advisor who is appointed by the chair for two to three year periods. The Lead Undergraduate Advisor works directly with the Undergraduate Faculty Advisor and supervises the other two staff members that make up the three-person unit.

The lead advisor bears the bulk of administrative duties and is responsible for the day-to-day management of the undergraduate advising staff members. All three have extensive interactions with our majors as well as with prospective students (our advising staff is fielding an exponentially increasing number of inquiries regarding appropriate training and education for careers in computing and information technology), recruitment and outreach activities, and address issues related to the recruitment and retention of women and underrepresented minorities.

## 5. Department Constituencies and Feedback Loops

The department's constituencies, in order of importance, are:

- current students in our major,
- prospective students (both at the University, at Community Colleges, and in high school)
- employers of our students and/or the graduate schools our students will enter, and
- the citizens of Washington State.

Input from these constituencies reaches the department in various ways. Current students interact with faculty and advising staff through individual meetings and special events (such as our monthly undergraduate lunches and other sessions on specific topics such as research/graduate-school/job-hunt sponsored by the student chapter of the ACM technical society). Comments and suggestions reach the Undergraduate Faculty Advisor and, through that position, the Curriculum Committee, Executive Committee, and/or Chair as appropriate. The remedies or suggestions can reach the entire faculty for discussion if they are far-reaching enough. Finally, the faculty implements the solutions generated through this process. This is the principal feedback loop that affects curriculum organization and content. Students in all our classes are also provided with a collection of anonymous feedback web forms that are specific to a course or relating to the entire program and department. Feedback is solicited through these forms to encourage an unencumbered exchange of ideas and reactions to the material presented in class and the overall curriculum design. This is particularly helpful in making students feel that their voice can be heard, taken seriously, and have a positive effect.

Information about prospective students reaches the faculty primarily through the undergraduate advising unit 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. These follow a similar path through the Curriculum Committee to the faculty. 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 as well as and interact with K-12 schools and community colleges regarding their preparation of students.

Current and prospective employers are strongly encouraged to join our department's Industrial Affiliates Program. An annual meeting, attended by technical representatives of our affiliate companies is an important venue where they can meet and recruit our students in a setting where students have contact to technical staff rather than just human resources specialists. Comments from our affiliates regarding our students' preparation is fed back to the faculty as a whole by the Industrial Affiliates Program Committee and appropriate action can be initiated through the advising unit and/or the Curriculum Committee. Another form of feedback about employment is through alumni surveys. Former students are strongly encouraged to provide us information about how well prepared they were for their careers once they have a longer-term perspective. Feedback concerning the preparation of our students for graduate school is obtained through our External Advisory Committee as well as faculty colleagues at the top graduate schools. This follows a similar path to industry comments.

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.

Important web sites explaining the ABET process and other aspects of the department:

Main CSE web page	<a href="http://www.cs.washington.edu">www.cs.washington.edu</a>
ABET materials	<a href="http://www.cs.washington.edu/abet">www.cs.washington.edu/abet</a>
Information on all our educational programs	<a href="http://www.cs.washington.edu/education">www.cs.washington.edu/education</a>
Information for our current students	<a href="http://www.cs.washington.edu/education/ugrad/current/">www.cs.washington.edu/education/ugrad/current/</a>
Handbook for our undergraduates	<a href="http://www.cs.washington.edu/education/ugrad/handbook/CSEhandbook.pdf">www.cs.washington.edu/education/ugrad/handbook/CSEhandbook.pdf</a> (including mission statement on page 12)
Information for prospective students	<a href="http://www.cs.washington.edu/education/ugrad/prospective/">www.cs.washington.edu/education/ugrad/prospective/</a>
Dep't strategic plan	<a href="http://www.cs.washington.edu/homes/lazowska/chair/plan2001.pdf">www.cs.washington.edu/homes/lazowska/chair/plan2001.pdf</a>
10-year review documents (2001)	<a href="http://www.cs.washington.edu/homes/lazowska/review/">www.cs.washington.edu/homes/lazowska/review/</a>
Materials on our inaugural Brotman award for educational excellence	<a href="http://www.cs.washington.edu/education/brotman.html">www.cs.washington.edu/education/brotman.html</a>
Capstone videos for the last several years	<a href="http://www.cs.washington.edu/info/videos/">www.cs.washington.edu/info/videos/</a>

## 6. Examples of the Feedback Loops Working

The table below illustrates some examples of the department feedback loops at work. The highlights shown in this table are only a sampling of the results of a process that has been in place for the past 3 years. It is important to note that a large collection of concerns regarding flexibility in the Computer Engineering program, availability of more courses dealing with embedded systems issues, modernization of requirements, decrease in EE requirements, and increase in free electives were all implemented as part of the Computer Engineering expansion we began to propose in early 1996, refined in the two subsequent years while we began to prototype courses and were finally able to implement beginning in the Autumn of 1999.

Qtr	Source of Information	Concern	Response
Wi 98	Students - discussed in Executive Committee	The rotating enrollment priority, which for spring quarter leaves Sophomores as the last to register, although Engr Sophomores all need 142 as a pre-req to get into their majors.	Restricted registration for 142/143 to freshman, soph, and juniors during registration period I. 143 priority to students currently in 142.
Wi 98	Faculty and students in reflective statements and course evaluations	Lack of motivation for required senior technical writing course	Exploring linking capstone design experience with required senior technical writing course (TC 333) so that students can write about the project they are currently working on in their capstone and provide more varied documentation. This requires co-registration and it has proven a challenge to get students to do this in large enough numbers for a complete section of TC 333 (~25 students). Successful trials were run in Wi 99, Au 99, and Sp 01.
Au 98	Prospective students and their families	Top students have no guarantee of admission to the department once enrolled in UW.	Implementation of Early Decision allowing us to admit up to 10% of the incoming class directly out of high school. This has been expanded to 20% for the fall of 2001 and has been adopted by many departments throughout the campus and especially in the College of Engineering.
Wi 99	Students at ugrad lunch	Info about undergrad labs & systems is lacking	A list of all labs, rooms and systems available to undergrads, linked from the "Information for Current Undergraduate Students" page.
Wi 99	Students at ugrad lunch	Student suggestion venue on Web	Anonymous (as well as non-anonymous) feedback page created where students can address comments to the faculty coordinator for the undergraduate program and advising staff. This page also includes a list of other resources for students: faculty members, department chair, staff advisors, faculty program coordinator, ACM chapter, tech support staff, lab director, etc. It has been useful so far in providing feedback on instructors, labs, and course scheduling.

Qtr	Source of Information	Concern	Response
Wi 99	Students at ugrad lunch	ACM in formal liaison role	The officers of the ACM (our student technical society chapter) met with the undergraduate faculty advisor at a long breakfast meeting. The students expressed a strong interest in taking a more active role as liaison between students and faculty/staff. ACM now coordinates bi-annual research nights for undergraduates to learn about research in the department as well as annual job-hunt preparation events among many others.
Wi 99	Students at ugrad lunch	Student exposure to heterogeneity	Curriculum Committee is going to ensure students become familiar with both C++ and Java and Unix and Windows as part of switch of introductory courses to Java.
Wi 99	Students at ugrad lunch	Ensure Unix environment is always available for students	All good X-terminals are being kept in service and are now split over two labs to make access easier. Windows software (Reflection) has been obtained and is now configured to take over the entire screen of Windows machines and lets students more familiar with Unix have a comfortable desktop for their needs.
Sp 99	Students at ugrad lunch	Escalate student concerns within Lab staff	Lab staff have been told that they need to route academic questions to faculty, and to escalate student concerns promptly that they think might be getting out of hand so that lab management and/or faculty can take action quickly.
Sp 99	Graduating students exit survey	Provide training in use of Unix platforms	Working with the advising unit, the ACM developed a set of short tutorials now held bi-annually to help students over the Unix learning curve.
Au 99	Graduate survey	Not easy to keep in touch with other alumni	Creation of an alumni webpage that includes links to alumni e-mail and webpages, etc.
Au 99	Exit survey	Increase flexibility so that students can take more foreign language credits	With the program expansion, additional free electives were provided to support a foreign language.
Wi00	Feedback from students/faculty	Appropriateness of EE332 (bipolar transistors) as a required course	The course was dropped from Computer Engineering requirements as part of expansion of program and institution of two curriculum options.

Qtr	Source of Information	Concern	Response
Wi00	Faculty reflective statement	CSE461 (networking course) should have a data structures pre-requisite	After curriculum committee review, CSE143 is added to course prerequisites so as to ensure some exposure to data structures without adversely impacting students from other departments (esp. EE).
Sp 00	Students	Too much paper used for course handouts	Several faculty surveyed classes and determined that although handout are overwhelming popular, more material could be fit on a page to decrease paper use.
Sp 00	Student exit survey	CSE 461 (networking) should be an approved outer core elective	The curriculum committee recommended that this change be made and the faculty approved it. Also, 461 is now a required course for all CompE majors.
Au 00	Students	Want more active role in outreach to incoming freshman, women, and minorities	Advisors recruited students at undergraduate lunches and asked others students to get involved. Over 20 students have signed up to assist with various recruitment and outreach activities.
Au 00	Students	Increasing number of inquiries about graduate school	Conducted first graduate school information night 10/00. Attended by over 60 students. Discussed selection process and preparation of applications.
Au 00	Students	Many courses in Applied Math are equivalent to some in Math required by the program	After checking with the Math department, the curriculum committee voted to allow substitutions from Amath for select required courses. At the same time, alternate Math courses were also included.
Au 00	Students and faculty	Undergraduates require a larger space (currently less than 250 sq ft) where they can work on research and independent projects	The Department through grants and gifts obtained by several faculty was able (through the University) to rent and remodel an office campus space of 3500 sq ft where a variety of student projects can be housed. This facility came on line during Spring 2001. See <a href="http://point5.cs.washington.edu">point5.cs.washington.edu</a> .
Wi01	Faculty reflective statement	CSE461 needs a statistics prerequisite	STAT390 (for CSE students) and IE315 (for EE students) are proposed as prerequisites and should have minimal impact on course scheduling for students.
Sp01	Students and faculty	Clarification needed for how research credit can be classified and if students can be paid	A web page was developed to cover all issues related to research participation and the dep't executive committee decided to permit pay for projects for which students are also receiving credit.

## **B. Accreditation Summary**

### **1. Students**

The department operates an Undergraduate Advising Office with three full-time professional advisors (Jennifer Seller, lead advisor; Elizabeth Rowson, advisor; and Crystal Eney, assistant advisor). The Undergraduate Faculty Advisor (Gaetano Borriello) supervises the overall advising effort. The office provides both pre-major advising to prospective students and advising and curriculum planning to current majors. A sample of the CSE Handbook is included in an appendix and at [www.cs.washington.edu/education/handbook/](http://www.cs.washington.edu/education/handbook/). 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.

Until recently, there were two separate advisors for the computer engineering and the computer science majors with a third support staff position. However, in 1997, following a turnover in the advising staff, it was decided that the best way to serve our entire undergraduate population was to have all three positions be advisors available to all CSE students. Now the three advisors are knowledgeable about both degree programs and can advise any CSE student rather than requiring students to meet with a single specific individual for all their advising needs. We also integrated the computer engineering and the computer science handbooks to ensure that all CSE students are receiving consistent and complete information about all the options available to them from the department.

#### Admissions

The Undergraduate Advising Office coordinates the departmental undergraduate admissions effort. Admission to the department is highly competitive. Students apply to the program after completing a set of program prerequisites and are evaluated based upon, but not limited to, the following criteria: grades in prerequisite and general education courses, a written personal statement, and potential to contribute to the field of computing. An Admissions Committee, comprised of four faculty members (including the undergraduate faculty advisor) and all four advising staff members, evaluate the applications. Three readers (two faculty members and one advisor) read each application prior to the committee meeting. The committee makes all final decisions by consensus. The past three years, the department has also guaranteed admission to highly competitive entering freshmen that have specified an interest in the department on their application to the UW. These students are not coded as majors until they complete the prerequisites and have maintained a 3.5 GPA in all their courses. All departmental admissions forms are on-line and students can complete the process remotely. Once the on-line application is completed, students print the information, attach their transcripts, and sign a statement that is accurate to the best of their knowledge. The completed packet must arrive at the department in time for two annual deadlines (one on July 1 for Autumn admission, one on Feb 1 for Spring admission).

Upon entry to the major, each student is provided a current copy of the 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 advisors also thoroughly present the curriculum and satisfactory progress policy at mandatory new-student orientations. Early Decision students receive a special orientation session tailored to their needs before starting their first fall quarter. 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.

### 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.

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 courses in the major, a faculty member who has recently taught the course in question, evaluates a packet of material provided by the student (for each course) that includes textbooks, assignments and the student's solutions, as well as syllabi provided by the instructor. The two courses that most often fall in 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](http://www.cs.washington.edu/outreach/CC/cc_transfer_application.html)). We work closely with community college instructors at annual summer workshops to ensure their introductory programming courses are well-aligned with ours and we have provided material ranging from lecture notes and assignment to complete sets of studio-quality video-taped lectures. We are currently actively engaged with the community college regarding the switch in programming language we are beginning to implement this Spring (C/C++ to Java).

In reality, the department admits only a handful of direct transfer students each year. Most of these are from other four-year institutions. Virtually all students who start their degrees at community colleges enroll at the UW campus for at least one quarter and thus do not appear on statistical summaries as direct transfers. Approximately 50% of our students have at least 5 credits that they have completed at institutions other than UW. But few have more than a fourth of their credits from other institutions by the time they graduate. Under the state Higher Education Coordinator Board agreement, students may transfer a maximum of 90 credits towards their degree. In all cases, the final 45 credits of the degree program must be earned while in residence at the University of Washington.

### Student monitoring

The staff maintains an interactive 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 is available to faculty as well and includes complete student academic records (linked to the University's main student database), a photograph, and comments entered by the advising staff after every meeting with the student. This allows us to keep a trail of interactions with each student that is accessible to all advising staff members and faculty.

At the end of each quarter, the advisors review the student 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 Faculty Advisor, students on probation for two consecutive quarters are transferred out of the program. This occurs only 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 tutoring program that is organized and staffed by current graduate and undergraduate students. Tutoring may include individual tutoring, directly by the student volunteers, or help in forming a study group with other CSE majors.

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 is at two to 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 includes a plan of courses in progress and to be scheduled through the quarter of graduation. The outcome is that each student is required to have an acceptable terminal course plan over two quarters in advance of actual graduation.

Exceptions and substitutions regarding prerequisites or degree requirements are only rarely granted, and then only with the explicit permission of the Undergraduate Faculty Advisor. If a student has taken classes at another four-year institution and wishes to transfer credit, they must file a petition. In addition to the petition, they submit copies of the syllabus, homework assignments and exams and name of text/s used. Faculty who teach the course for which credit is being requested, review the petition and accompanying documents and make the final determination on whether transfer credit should be granted.

### 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 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 a semi-annual orientation events that coincides 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. In both cases, students work closely with faculty (and their graduate students) to learn more about other aspects of the department than their coursework.

## 2. Program Educational Objectives

### Program Mission Statement

**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](http://www.washington.edu/faculty/facsenate/handbook), the mission of the College of Engineering is available at: [www.engr.washington.edu](http://www.engr.washington.edu) ).

From our program mission statement we derive the following four goals for our Computer Engineering program:

- Core knowledge: To offer a challenging and broad-based education in the fundamental ideas, techniques, and tools underlying the discipline of computer engineering.
- Application: To impart the ability to apply the core knowledge of the discipline in making the many engineering tradeoffs that are central to solving problems and creating products.
- Educational renewal: To prepare our students to continually renew their education in a rapidly developing discipline so that they may realize their full potential throughout their career.
- Social context: To foster personal development not only in the social aspects of the discipline itself but also in terms of understanding the impact of our field on society as a whole.

We believe these four goals clearly organize the department's educational values, namely, graduating highly competent students who can serve as leaders of the field for their entire careers and do so understanding the implications of their work both to themselves and society as a whole.

We further refine these four high-level goals into 17 objectives. Table I is organized with these objectives under their respective goals. In addition, it also demonstrates how the objectives taken together address all 11 of the ABET educational outcomes criteria (i.e., "A-to-K").

### Constituencies

A discussion of the department's educational program constituents was already presented in section A.5. The list can be further refined into primary, major, and minor constituencies. Of course, feedback from any of these constituents becomes especially compelling when it correlates with that of other constituent groups.

Our primary constituents are:

- students currently enrolled in our major,
- students seeking to become majors in our programs,
- and past graduates of our programs.

Major constituents are composed of those organizations serving our student's careers immediately after they leave UW, namely:

- our external advisory committee,
- member companies of our industrial affiliates program and other prospective employers, and



- graduate school faculty at the nation's premier graduate programs.

Finally, the largest, and most diffuse, constituents are:

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

### Process

Initially goals and objectives were derived by the Undergraduate Program Coordinator (then also serving as Associate Chair for Educational Programs) and the advising staff with the help of numerous discussions with the College's ABET Coordinators Committee (on which the Undergraduate Program Coordinator and Lead Undergraduate Advisor serve). These were then presented to the Curriculum Committee (which includes two undergraduate students, two graduate students, four faculty, and two advisors) and were revised extensively. Finally, they were brought to the entire faculty at a regular faculty meeting and at annual faculty retreats. Presentations by the Undergraduate Program Coordinator explained the reasoning behind the goals and objectives. Extensive feedback was received that led to a second major revision and the result is shown in Table I. These are now posted on the department web pages and are gathering additional comments from students, affiliate companies, and faculty in other departments. We fully expect further refinements to be made on a regular basis.

Some examples of the how feedback from various groups has led to changes in our program and the service we provide our students are described in Section A.6.

### Achievement

The educational objectives are realized through careful curriculum design, continual monitoring of students' progress, assessment of outcomes, and evaluation of the curriculum by our primary and major constituencies.

The curriculum structure and its rationale is described in detail in Section B.3 and B.4 and consists of components relating to core engineering, science, and mathematics knowledge, the fundamentals of the discipline, and, depending on the student's chosen program option, a set of advanced courses that put the fundamentals to practice. A capstone design course completes the curriculum with an open-ended individualized design experience. 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.

Core courses in the department's curriculum are offered quarterly to make it easier for students to complete their schedules. Careful attention is given to course scheduling to minimize conflicts among our own offerings as well as those of related departments such as Electrical Engineering and Technical Communications.

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-ware, future plans for the course content and assignments, and how previously identified issues were handled. In addition, the Undergraduate Advising Office and the Undergraduate Program Advisor are continually in contact with our primary and major constituencies and bring issues that merit attention to the Curriculum Committee.

When deficiencies or other issues are identified the Undergraduate Program Advisor and the Advising Office work with the relevant faculty to produce a workable solution and, if needed, bring it before the Curriculum Committee and possibly the entire faculty. Again, the reader is referred to Section A.6 for specific examples.

Table I. Goals (1-4) and objectives (1.1-4.4) and their mappings to the ABET criteria (A-K).

	A	B	C	D	E	F	G	H	I	J	K
<b>1. Core knowledge: To offer a challenging and broad-based education in the fundamental ideas, techniques, and tools underlying the discipline of computer engineering.</b>											
1.1 Apply a broad-based knowledge of math, science, and computing fundamentals to identify, formulate, and solve computer engineering problems.	X										
1.2 Apply the fundamentals of analysis, synthesis, abstraction, simulation, and experimentation to design and implement solutions to challenging problems.		X									
1.3 Convey technical concepts through an appropriately targeted oral presentation and interaction with an audience.							X				
1.4 Convey technical concepts through written materials that satisfy accepted standards for writing style.							X				
<b>2. Application: To impart the ability to apply the core knowledge of the discipline in making the many engineering tradeoffs that are central to solving problems and creating products.</b>											
2.1 Decompose design problems and choose appropriate components for their solution.	X		X								
2.2 Assemble and integrate software and/or hardware components into new products.		X	X		X						
2.3 Apply the techniques and tools of the discipline in the evaluation of design tradeoffs.		X			X						X
2.4 Function effectively on a multi-disciplinary team.				X			X				X
2.5 Participate in a range of learning opportunities including large implementation projects, research apprenticeships, teaching assistantships, and independent study.							X		X		X
<b>3. Educational renewal: To prepare our students to continually renew their education in a rapidly developing discipline so that they may realize their full potential throughout their career.</b>											
3.1 Recognize the need for, and will have practiced, self-directed learning.									X		X
3.2 Recognize the need for, and be able to engage in, life-long learning.									X		
3.3 Understand career options that lead toward graduate school and/or industry practice.									X		
3.4 Understand the variety of disciplines and industries to which computer engineers can contribute.										X	
<b>4. Social context: To foster personal development not only in the social aspects of the discipline itself but also in terms of understanding the impact of our field on society as a whole.</b>											
4.1 Foster a collaborative environment where students, staff, and faculty are all engaged in the continual improvement of the department.						X		X			
4.2 Appreciate the complementary roles individuals of different background can play on engineering teams.			X								
4.3 Understand the ethical issues facing computing professionals and awareness of resources available to those faced with ethical decisions.						X				X	
4.4 Knowledge of historical and contemporary issues and the impact of computer engineering solutions in a global and societal context.								X		X	X

### 3. Program Outcomes and Assessment

#### Program Outcomes

We view program outcomes as measurable effects of our curriculum. The particular choice of outcomes was strongly influenced by our program objectives. As such, there is close correspondence between them.

Our program outcomes are (grouped loosely into four groups corresponding to the program goals):

- 1.a Mastered mathematics and science fundamentals.**
- 1.b Applied mathematics and science concepts.**
- 1.c Presented an effective oral presentation.**
- 1.d Prepared effective written materials.**
  
- 2.a Mastered core of the discipline of computer engineering.**
- 2.b Applied core concepts of the discipline to design/evaluation.**
- 2.c Constructs appropriate abstractions to manage complexity.**
- 2.d Able to evaluate design tradeoffs.**
- 2.e Completed a challenging design project.**
  
- 3.a Facile in using tools of the discipline.**
- 3.b Worked on a multi-person design team.**
- 3.c Participated in diverse learning opportunities.**
- 3.d Assimilated knowledge of tools or concepts as needed.**
  
- 4.a Interacted with industry practitioners.**
- 4.b Considered benefits of graduate study.**
- 4.c Understands principal concepts of professional ethics.**
- 4.d Evaluated implications of work in the discipline to society as a whole.**

The outcomes are derived from our objectives and represent quantifiers that are relevant to each of the objectives. As such, they have almost a one-to-one correspondence with the program objectives laid out in Table I, and we do not provide a table showing this direct correspondence.

#### Relation to Program Educational Objectives and ABET Criteria

By virtue of corresponding directly to the program objectives, the outcomes also cover the ABET criteria (A-K) in a similar manner.

The criteria specific to a Computer Engineering Program include: diversity of faculty; their active participation in research in the discipline; curriculum coverage for a balanced view of hardware, software, hardware-software tradeoffs, and basic modeling techniques used to represent the computing process; breadth of program offerings; required depth in a topic area with a sequence of courses that build upon each other.

The department's faculty covers the discipline handily and all its members are active participants in the research enterprise. The department is ranked among the top 10 in the US in computer science and engineering (and almost always in the top 5 or 6) based primarily on the stature of its faculty and its research and educational impact. Our Department, College, and University tenure committees ensure that this is the case.

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 with particular emphasis in the context of embedded systems. A set of common core courses forms a foundation on which senior requirements are layered. Appropriate capstone design courses are

designated for the two options. Senior-elective requirements ensure breadth outside the area of specialization.

### Processes

Outcomes are achieved via both curricular and extra-curricular activities. The curriculum is designed to provide students with a solid foundation and extensive practice in an area or two of specialization. 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 with 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.

In terms of assessment, in addition to course grades, we focus on the development of portfolios based on students' work in senior-level project-oriented courses and in capstone design courses. Web-based documentation, oral presentations, and video productions serve to populate the portfolio. The material comes from departmental courses as well as the required technical communication courses (the last of which we are working to link with our capstone design courses so that students enroll in both concurrently and can write about their capstone project in the technical writing courses – providing a stronger motivation).

The audience for student portfolios is quite broad and includes faculty, advising staff, and prospective employers. 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 (they are broadcast on the UW cable television channel) and prospective students. Comments from all these groups (some as formal surveys and others as informal observations) are gathered and continually considered by the Undergraduate Faculty Advisor and the Advising Office for possible action.

Students evaluate ALL our courses each quarter. 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.

Faculty members also evaluate the courses they teach by filing 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 is collected and overall advice developed to help the faculty member improve their performance.

The Undergraduate Program Advisor 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 Advisor to discuss issues that have arisen. The Curriculum Committee meets once or twice per month to discuss curriculum changes as well as issues forwarded by the Undergraduate Advising Office that are likely to require further attention and a possible faculty vote. Examples of this include everything from minor changes to the approved elective list to proposals to create a new option for the major.

Finally, both the department and the Office of Educational Assessment (OEA) conduct surveys of alumni. The department maintains an alumni web site where students keep their current status up to date and can send feedback to former professors. The focus is on particularly memorable

experiences while they were students and how prepared they felt/feel on the job. The OEA surveys are conducted of students both 1 and 5 years past graduation and include several questions that can be mapped to the program outcomes.

### Curricular Coverage of Outcomes

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

Table II illustrates how they are used to obtain our program outcomes. Each of these bears some explanation. General education and VLPA/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, differential equations, and linear algebra. A statistics course for scientists and engineers is also required. Science requirements are dominated by a year of physics and corresponding laboratories with one quarter for exposure to introductory chemistry. Students can further increase their range using their free electives. Many of our students pursue a foreign language 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 University 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 8 required courses. Students can choose between two options for outer core requirements: one focusing on hardware systems and the other on embedded software. Each option has three required senior level courses. In addition, a capstone course is required and students have a choice of one of two different capstones for each option. The capstone courses tie 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.

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 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 50% 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 25% of our students participate in research with faculty and graduate students. They can often earn a salary for this work letting them spend more time on

campus while earning an income. Finally, approximately 10% of our graduating class each year has participated in teaching assistantships 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 Faculty Advisor 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 had the experiences.

Table III provides a summary of our curriculum requirements and the mapping to our outcomes.

### Assessment

A summary of our assessment process can be found in Table IV. For each of our outcomes, we provide performance criteria, the implementation plan, evaluation methods, logistics involved in collecting the required information for evaluation, and who provides the feedback to the student. A meta-feedback loop on all of these is that the faculty are the first line for detecting issues that need attention. Their comments are provided to the advising staff and Undergraduate Faculty Advisor directly and indirectly through course grades and self-evaluations of teaching. Student feedback is provided through course evaluations, anonymous web forms, and public forums of discussion such as the undergraduate lunches held monthly by the Chair and attended by other faculty as well. Many issues can be handled directly by the advising staff. More systemic or problematic issues are forwarded to the Undergraduate Faculty Advisor in weekly meeting with the Undergraduate Advising Office staff. Next in line is 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 bureaucratically necessary for a vote and, more importantly, as part of a continual faculty education process.

The department uses several forms of assessment:

- Course grades
- Course student evaluations
- Faculty self-evaluations
- Student surveys upon entry and exit from the department
- Surveys related to industry employment (pre/post and employer) through the College of Engineering's Coop Office
- Capstone design projects and video productions
- Writing assessment through the Department of Technical Communications
- Presentations (posters, demos) at the annual affiliates meeting
- Alumni surveys

The combination of all of these metrics provides the leadership of the department (and the faculty) with a multi-faceted picture of the relative strength of the curriculum as seen from the point of view of most of our constituents.

Table II. Program outcomes and mapping to curricular and extra-curricular elements.

	Curriculum Component (credits)										Extra-curricular		
	General education (42)	VLP/PA/IS (30)	Communication (12)	Math and statistics (25)	Science (20)	CSE core (32)	HW/SW outer core (16-18)	CSE electives (13)	Free electives (13-15)	Industry internships	Research project	Teaching assistantships	
<b>1. Core knowledge:</b>													
1.a Mastered mathematics and science fundamentals.				X	X								
1.b Applied mathematics and science concepts.					X	X	X						
1.c Presented an effective oral presentation.			X				X						
1.d Prepared effective written materials.			X				X						
<b>2. Application:</b>													
2.a Mastered core of the discipline of computer engineering.						X	X						
2.b Applied core concepts of the discipline to design/evaluation.							X						
2.c Constructs appropriate abstractions to manage complexity.						X	X						
2.d Able to evaluate design tradeoffs.							X						
2.e Completed a challenging design project.							X						
<b>3. Educational renewal:</b>													
3.a Facile in using tools of the discipline.						X	X	X					
3.b Worked on a multi-person design team.							X			X			
3.c Participated in diverse learning opportunities.	X	X					X	X	X	X	X		
3.d Assimilated knowledge of tools or concepts as needed.							X			X			
<b>4. Social context:</b>													
4.a Interacted with industry practitioners.										X			
4.b Considered benefits of graduate study.										X	X	X	
4.c Understands principal concepts of professional ethics.	X	X	X				X						
4.d Evaluated implications of work in the discipline to society as a whole.	X	X					X			X			



Table III. Program outcomes and mapping to curriculum requirements.

		1.a	1.b	1.c	1.d	2.a	2.b	2.c	2.d	2.e	3.a	3.b	3.c	3.d	4.a	4.b	4.c	4.d
<b>Core Courses</b>																		
	Mathematics		X															
	Science		X															
	General Education/VLPA-IS																X	X
	Communication			X	X												X	X
CSE 142	Computer Programming I					X					X							
CSE 143	Computer Programming II					X					X							
CSE 321	Discrete Structures		X			X					X							
CSE 322	Intro. to Formal Models					X					X							
CSE 326	Data Structures					X		X			X							
CSE 341	Programming Languages					X					X							
CSE 370	Intro. to Digital Design		X			X		X			X							
CSE 378	Machine Org. & Assembly Lang.					X					X							
CSE 451	Operating Systems					X	X	X	X		X							
CSE 461	Networking					X	X	X	X		X							
EE 215	Fund. of Elect. Engineering	X	X			X												
EE 233	Circuit Theory		X			X					X							
<b>Hardware Option</b>																		
CSE 467	Advanced Digital Design							X	X	X	X	X						
CSE 471	Computer Design & Org.							X	X	X		X						
EE 331	Devices & Circuits I		X			X					X							
<b>Capstone Design</b> One of the following:																		
CSE 468	Very Large Scale Integration		X	X	X	X	X	X	X	X	X	X		X	X			
CSE 477	Digital Systems Design			X	X	X	X	X	X	X	X	X		X	X			
<b>Software Option</b>																		
CSE 403	Software Engineering							X	X	X	X	X			X			
CSE 466	Software for Embedded Sys.							X	X	X	X	X						
CSE 401/457/471	Senior Project oriented course							X	X	X	X							
<b>Capstone Design</b> One of the following:																		
CSE 476	Embedded Systems			X	X	X	X	X	X	X	X	X		X	X			
CSE 481	Software Design			X		X	X	X	X	X	X	X		X	X			
<b>Senior Electives</b> Also any courses listed above																		
CSE 421	Intro. to Algorithms							X	X	X	X							
CSE 431	Intro. to Theory of Computation		X					X			X							
CSE 444	Databases							X	X	X	X							
CSE 458	Computer Animation							X	X	X	X	X			X			
CSE 473	Intro. to Artificial Intelligence							X	X	X	X							
CSE 498/499	Senior Project/Research							X	X	X	X		X			X		

Table IV. Assessment guide for departmental outcomes.

**Goal 1. Core knowledge: To offer a challenging and broad-based education in the fundamental ideas, techniques, and tools underlying the discipline of computer engineering.**

Program Outcomes	Performance Criteria	Implementation Plan	Evaluation Methods	Logistics	Feedback
1.a Mastered mathematics and science fundamentals.	Ability to solve problems that arise in core computer engineering courses.	Core courses in mathematics, statistics, physics, chemistry, and computing.	Course grades and sample assignments.	Grades reviewed by Undergraduate Advising Office.	Lead Advisor and Undergraduate Faculty Advisor raise issues of concern with Curriculum Committee.
1.b Applied mathematics and science concepts.	Ability to solve problems that arise in all computer engineering courses.	Senior-level courses as well as independent projects and research with faculty supervision.	Course grades, sample assignments, and project documentation.	Collected at the end of a course or project by the Advising Office. Each capstone course produces a video of a selection of projects.	Project complexity judged by faculty and industrial affiliates.
1.c Presented an effective oral presentation.	Ability to prepare and deliver an oral presentation on a capstone or independent project.	Capstone design courses and technical communication requirements.	Self-evaluation, peer-evaluation, and instructor evaluation of presentation.	Video-taping in technical communications courses.	Technical communication instructor provides direct feedback to student.
1.d Prepared effective written materials.	Project documentation that can be used by other students and faculty to evaluate and build on to project.	Term papers in senior courses. Capstone design course documentation and technical communication requirements.	Cross-presentations prepared using materials from other project group.	Web-based project documentation archived for review by faculty and future students.	Fellow students use the material to build on to projects in future courses.

**Goal 2. Application: To impart the ability to apply the core knowledge of the discipline in making the many engineering tradeoffs that are central to solving problems and creating products.**

Program Outcomes	Performance Criteria	Implementation Plan	Evaluation Methods	Logistics	Feedback
2.a Mastered core of the discipline of computer engineering.	Ability to solve problems that arise in senior-level courses using concepts from core courses.	Core and senior-level courses in CSE.	Course grades and sample assignments.	Grades reviewed by Undergraduate Advising Office.	Lead Advisor and Undergraduate Faculty Advisor raise issues of concern with Curriculum Committee.
2.b Applied core concepts of the discipline to design/evaluation.	Completion of challenging projects in senior-level courses and in capstone design courses.	Core and senior-level courses in CSE.	Course grades and project assignments.	Faculty evaluation of courses.	Project complexity judged by faculty peers who teach the same courses.
2.c Construct appropriate abstractions to manage complexity.	Developed abstractions as part of the process of solving problems and implementing ideas in all courses that emphasize design.	Capstone design courses and independent projects.	Course grades and sample assignments. Project documentation.	Grades reviewed by Undergraduate Advising Office. Documentation reviewed by faculty.	Project complexity judged by faculty peers who teach the same courses.
2.d Able to evaluate design tradeoffs.	Challenging projects in capstone design courses and ability to relate the thought process that went into the tradeoffs.	Capstone design courses.	Presentations and documentation in capstone design courses.	Presentation to peers with specific requirements to highlight tradeoffs and reasoning behind them.	Faculty provide feedback through evaluations in design courses.
2.e Completed a challenging design project.	Successful demonstration as part of capstone design course including video produced by UWTV.	Capstone design and research assistantship opportunities.	Demonstration at end of capstone design experience. Participation in undergraduate research seminar.	Presentation to entire department at graduation ceremony through video production. Poster session at industrial affiliates meeting.	Peers, faculty, and industry affiliates judge project relevance and comparison to state-of-the-art.

**Goal 3. Educational renewal: To prepare our students to continually renew their education in a rapidly developing discipline so that they may realize their full potential throughout their career.**

Program Outcomes	Performance Criteria	Implementation Plan	Evaluation Methods	Logistics	Feedback
3.a Facile in using tools of the discipline.	Ability to solve problems that arise in senior-level courses using concepts from core courses.	Project-oriented courses, capstone design, independent projects.	Course grades and sample assignments.	Grades reviewed by Undergraduate Advising Office.	Lead Advisor and Undergraduate Faculty Advisor raise issues of concern with Curriculum Committee.
3.b Worked on a multi-person design team.	Capstone design experience guarantees all students participate in group work.	Capstone courses and independent projects.	Presentations and demonstrations of work.	Presentation to entire department at graduation ceremony through video production. Poster session at industrial affiliates meeting.	Directly from faculty and/or TAs managing and meeting with groups in capstone design or project courses.
3.c Participated in diverse learning opportunities.	Participation in industry internships, independent study/research with faculty, and/or teaching assistantship on campus.	Career development activities by Industrial Affiliates Program, seminars organized by ACM chapter, career preparation workshops.	Coop program evaluation, poster sessions at affiliates meeting, undergraduate research seminar, student evaluations of TAs.	Keep track of participation in student database. Collect project reports for faculty review.	Directly from faculty involved in supervision (whether research or teaching). Post-internship surveys of students and employers.
3.d Assimilated knowledge of tools or concepts as needed.	Challenging capstone design experience.	Contact with member companies of Industrial Affiliates Program and seminars organized by ACM chapter.	Completion of design project with minimal guidance.	Successful demonstration at end of capstone design course.	Peers, faculty, and industry affiliates judge project relevance and comparison to state-of-the-art.

**Goal 4. Social context: To foster personal development not only in the social aspects of the discipline itself but also in terms of understanding the impact of our field on society as a whole.**

Program Outcomes	Performance Criteria	Implementation Plan	Evaluation Methods	Logistics	Feedback
4.a Interacted with industry practitioners.	Completed industry internship. Poster at industrial affiliates meeting. Attendance at department colloquia and ACM-sponsored industry speaker events.	Chair, faculty, and advising staff leadership. Presentations at orientations.	Participation at annual affiliates meeting. Internship participation rates.	Coop and Placement office records, Advising Office records in CSE database, observation at affiliates meeting.	Through ACM sponsored events regarding job search and career choices.
4.b Considered benefits of graduate study.	Participation and/or inquiries about independent study/research.	Bi-annual research nights to expose students to research in department.	Participation in independent study/research. Involvement of our graduate students with undergraduates.	Advising Office records all participation in independent study/research and directs students to appropriate faculty for graduate school advice.	Through ACM sponsored events regarding graduate school options.
4.c Understands principal concepts of professional ethics.	Ability to discuss issues, if not solutions, specific to computing technologies.	Ethics and society course content, ethics and society seminar, written analysis of societal impact of capstone projects.	Critique in technical writing courses coupled with capstone courses.	Part of web-based project documentation.	Directly by technical communications and CSE faculty in coupled courses.
4.d Evaluated implications of work in the discipline to society as a whole.	Written analysis of capstone project with respect to societal impact.	VLPA requirements, ethics and society seminar, department colloquia, ACM-sponsored presentations.	Critique in technical writing courses coupled with capstone courses.	Part of web-based project documentation.	Directly by technical communications and CSE faculty in coupled courses.

Assessment (continued)

**Course grades:** The Undergraduate Advising Office reviews all grades each quarter and identifies students that are not making satisfactory progress towards the degree. These students are called in for an appointment with an advisor to discuss what measures they should take to remedy the situation. The department has a volunteer graduate student-run tutoring program that provides direct help to students or helps them form study groups. Course teaching assistant also perform this latter role in ensuring their students are working smart and benefiting from each other without over-stepping bounds on collaboration established for the course. 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 3.7.

**Course student evaluations:** Every course in the department is formally evaluated by students. Anonymized histograms of the results for faculty and teaching assistants are broadcast by the Chair to the entire department population. Results over the past several years, since this policy was instituted, have been steadily improving. An average evaluation five years ago was 3.7-3.8 (out of 5) and today is closer to 4.2.

**Faculty self-evaluations:** Every time a faculty member completes a course they must file a self-evaluation that discusses what worked well and what didn't both for the instructor specifically as well as the material of the course. This reflective writing exercise provides invaluable material for future instructors.

**Student surveys upon entry and exit from the department:** These surveys include the names of students so that we can track the difference when they complete the program. Students are asked to rate themselves on the following 15 questions:

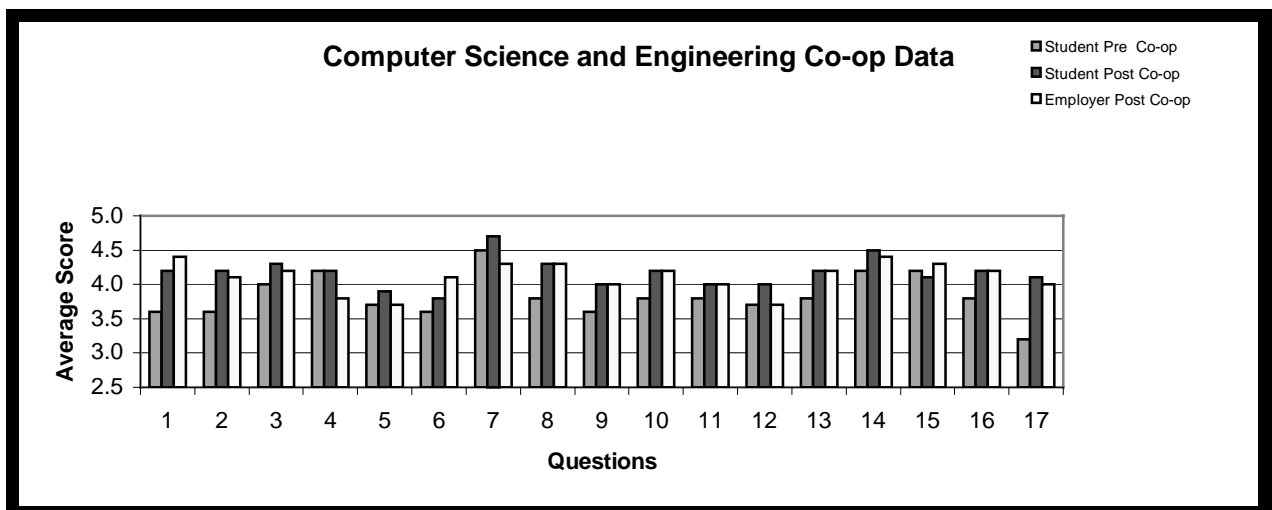
1. Ability to apply engineering theory and principles to the work environment.
2. Ability to design a system, component, or process to meet desired needs.
3. Ability to communicate effectively: on an interpersonal level; in formal presentations; in technical writing
4. An understanding of professional and ethical responsibility.
5. Recognize the need for lifelong learning.
6. Ability to use techniques, skills and modern engineering tools necessary for engr practice.
7. Ability to design and conduct experiments.
8. Ability to analyze and interpret data.
9. Ability to function on multi-disciplinary teams.
10. A knowledge of contemporary issues in computing.
11. Ability to identify, formulate and solve engineering problems.
12. Recognize the value of hands-on experience.
13. Ability to think critically.
14. Ability to understand the implications of rapid technological change.
15. Understanding of the work place and work-related social dynamics.

We find that the most valuable aspects of this survey are not the specific answers students provide. As one can imagine, students are hardly prepared to answer the above questions accurately when they first enter the department as they are quite unsure as to their meaning. However, we do find that it sets student's expectations for what their educational experience should provide and they begin to stop thinking of their courses as obstacles to surmount. It also provides a context in which they can make suggestions or lodge complaints about their experience in the department. As they leave the program, seeing the same questions again with their original answers (during their pre-graduation appointments) gives them an increased awareness of the issues they need to address in their comments (a listing of the last year's is available in the appendix). We do not present the numerical values since they provide little, if any, information due to the variance in interpretation among students and even by the same student at entry and exit.

Surveys related to industry employment (pre/post and employer) through the College of Engineering's Coop Office: CSE has worked with the Coop Office to develop surveys to be used by student before leaving for their internship/coop experience, upon their return, and by their employer after the student has returned to UW. 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.

1. apply engineering theory & principles to work
2. design system, component or process to meet desired need
3. professional and ethical responsibility
4. interpersonal communication skills
5. formal presentation communication skills
6. technical writing skills
7. recognize need for lifelong learning
8. use techniques, skills, and modern tools necessary for practice
9. design and conduct experiments
10. analyze and interpret data
11. function on multi-disciplinary teams
12. knowledge of contemporary issues
13. identify, formulate, and solve engineering problems
14. recognize value of hands-on experience
15. critical thinking
16. managing change
17. understanding office dynamics

The responses for the past year are summarized in the graphic below. Note that the primary areas of improvement are in questions 17, 1, 2, 13, and 8. These are precisely what we would hope to see. Unfortunately, industry does not involve interns in much multi-disciplinary activity. Second internships with the same company tend to show stronger results in that area. Interestingly, the only area where student actually feel worse-off after completing an internship is 15. This may be an indication of the approach taken by many companies in not giving large responsibilities to short-term workers.



**Capstone design projects and video productions:** We capture most of the projects from our capstone projects in video productions that highlight the projects and their utility and/or cool-factor as well as the students' own reflections on their projects. The department now has an extensive collection of these available on-line at: [www.cs.washington.edu/info/videos/](http://www.cs.washington.edu/info/videos/). The

videos are shown to our industrial affiliates, prospective students, parents at graduation ceremonies, and are broadcast on UWTV on a regular basis. They are professionally produced by UWTV and provide an important resource for the department's outreach efforts. In addition, they provide an excellent record of what our students were able to accomplish so that we can verify that our capstone design experience are state-of-the-art or a bit ahead of industry practice. We target our capstone projects to be where commercial products will be in a year or two after the student graduates. Many of our graduates have sent us e-mail when they discover a commercial product similar to their capstone design project. This projects a positive image of the department as being cognizant of the cutting edge and able to bring their students to practice at that level.

**Writing assessment through the Department of Technical Communications:** As part of the college-wide writing assessment program, two of our faculty (Kautz and Seitz), together with a faculty member from technical communications (Plumb), evaluated a random sample of senior-level Computer Engineering student papers during the Spring of this year. In order to improve inter-rater reliability, the three raters first looked at three "calibration papers," then met to discuss their ratings. Discussion at the meeting led to a better understanding of the three categories of Not Acceptable, Competent, and Strong. After the meeting, the three raters evaluated 16 other papers. Plumb evaluated all 16 papers, whereas Kautz and Seitz evaluated 8 papers each for a total of 32 ratings. No papers produced unacceptable differences among the raters (i.e., Not Acceptable vs. Competent or Not Acceptable vs. Strong). The papers used for the process were from TC 333 (Advanced Technical Writing and Oral Presentation). The TC 333 papers were written by individual Computer Engineering students and were randomly selected from the TC 333 papers from spring quarter 2000 through winter quarter 2001. The original three categories were Not Acceptable, Competent, and Strong. The Competent category was subsequently subdivided to include three categories: Low Competent, Competent, and High Competent. In order to get a Not Acceptable rating, a paper needed to show evidence that the student was not ready to write on the job. Of the 32 ratings, 4 were Not Acceptable, 3 were Low Competent, 17 were Competent, 7 were High Competent, and 1 was Strong. Two papers were rated Not Acceptable by both evaluators. In addition to this coarse-grained evaluation, the evaluation team considered the specific Performance Outcomes for UW Engineering Writing and noted when an individual outcome was weak. The five most frequently checked outcomes (thus the weakest), in order of most weak to less weak are listed below, with the number of times the outcome was checked:

- Uses proper citation form. (14)
- Includes citations for other's ideas, including any information and non-textual material from sources outside the writer. (13)
- Uses correct punctuation, grammar, usage, and spelling. (13)
- Content supports the purpose thoroughly and concisely. (10) Note: The evaluators discussed the problem of the duality in this outcome and the need to revise it. Most of these checks dealt with a lack of thoroughness rather than a lack of conciseness.
- Exhibits a logical progression and structures the content to represent that logical progression. (8)

In conclusion, the fact that 2 of the 16 papers (or 12.5%) were deemed Not Acceptable by both raters shows room for program improvement. The weak areas in these student papers spanned nearly all aspects of writing, from grammar and spelling to content and organization. Across the board, papers were weak in using sources when appropriate and citing sources using a conventional format.

In the short term, the recommendation is that more emphasis should be given (both in the stand-alone writing courses and in the departmental courses) to assigning writing that involves using sources. If the papers had shown consistent weaknesses primarily in content areas, these weaknesses could have, perhaps, been ascribed to a weak assignment or lack of motivation. However, the weak performance on the broad range of desired outcomes may call for more writing practice of various types in varied settings. The Director of the College's communication program



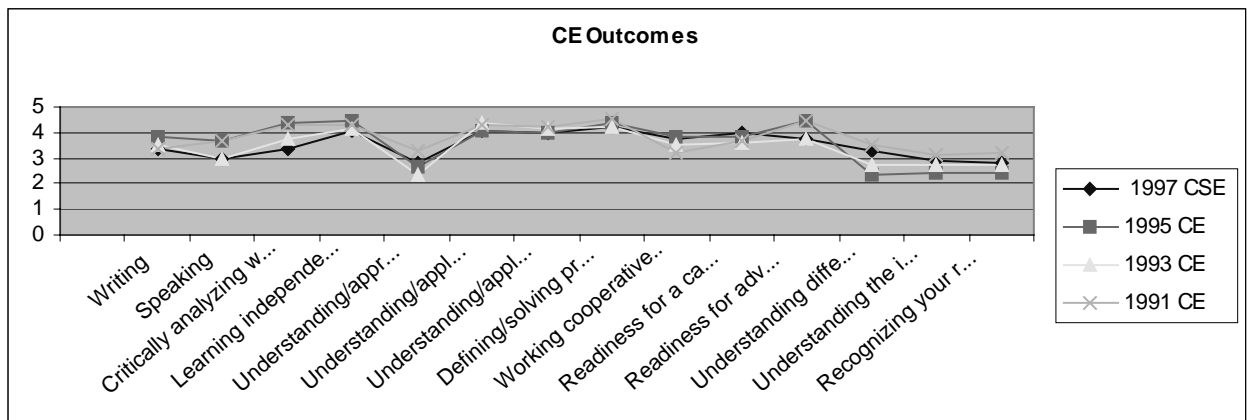
has agreed to work with CS&E to identify a senior-level CS&E course from which to sample student papers so that a wider variety of papers can be evaluated for the next assessment cycle. In the meantime, the department is also experimenting with linking capstone design courses with technical communication courses so that students can write more about their projects and be more motivated in their writing. It is too early to report on this experiment as it is currently in progress.

**Presentations (posters, demos) at the annual affiliates meeting:** Undergraduates participating in research are encouraged to prepare posters and demos for our annual industrial affiliates meeting. This gives them an opportunity to discuss their projects with leading industry researchers and receive valuable feedback. Participation rates at this event are steadily increasing among our undergraduates as they participate more heavily in research. Feedback from the affiliate members is usually provided directly to the faculty supervising the projects. Students in capstone design courses when the affiliate meeting takes place also prepare posters on their course projects.

**Alumni surveys:** Assessing how our students do when they leave our program is done through alumni feedback on our own alumni web pages as well as through surveys conducted by the Office of Educational Assessment. Graduates are asked to complete the survey 1 year after graduation. They do so with a close to 40% participation rate. Another survey is conducted after five years but this has too low a participation rate to be useful. Students are asked to rate themselves in the following 14 areas:

1. Writing
2. Speaking
3. Critically analyzing written info.
4. Learning independently
5. Understanding/appreciating the arts
6. Understanding/applying scientific principles/methods
7. Understanding/applying quantitative principles/methods
8. Defining/solving problems
9. Working cooperatively in a group
10. Readiness for a career
11. Readiness for advanced education
12. Understanding differing philosophies and cultures
13. Understanding the interaction of society and the environment
14. Recognizing your responsibilities, rights and privileges as a citizen

The survey is conducted every two years and the data for the past 4 surveyed classes is shown below.



The important messages from this summary is that our last ABET accreditation report was correct in its identification of ethics and societal impact as an area in which our curriculum is lacking. With the measures we are taking to remedy this situation (see section A.3), we hope to see these numbers change over coming years. Other low marks are for questions 5, which we do not directly address in our curriculum and do not plan to do so in the general sense of this question, and 2, which has led us to develop more opportunities for students to speak in front of their peers on technical concepts. Most of this is being done in the capstone design courses.

In addition to these OEA-sponsored surveys, the department conducts its own alumni survey, on a continuous basis, on our alumni web pages. We provide alumni with permanent e-mail addresses and a way to re-link with fellow students. This makes the pages highly visited. We collect broad comments using the following 3 questions as starting points:

1. Thinking back, is there a particular course(s) and/or instructor(s) who made a difference in your ability to succeed?
2. What did the department do well?
3. What could the department have done better?

The responses to these questions are available to all faculty to view and they are reminded of their existence on an annual basis. Comments are processed in the same way as our exit survey results and have lead to kudos to specific faculty as well as suggested changes to our curriculum committee.

#### *Process Results and Improvements*

The feedback loops described above are working. Section A.6 provides a three-page table outlining recent improvements made and how they were initiated. We do not reproduce it here. In addition, several improvements are suggested from the assessments described above in this section. In general, the ABET self-study process has proven an excellent experience for the department in spurring the development of additional assessment methods.

#### 4. Professional Component

The Computer Engineering program curriculum is described in detail in the department undergraduate programs handbook which is appended to this self-study in its own binding. It can also be found on the web at:

[www.cs.washington.edu/education/ugrad/handbook/CSEhandbook.pdf](http://www.cs.washington.edu/education/ugrad/handbook/CSEhandbook.pdf). The curriculum is designed by faculty as a whole through the Curriculum Committee and the Undergraduate Faculty 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. It is not repeated here in its entirety for the sake of brevity.

In summary, the program consists of the following requirements based on a 180 credit degree program under a quarter system. There are two options based on hardware or software emphasis in embedded computing systems. All ABET requirements are met as can be seen in Table A.1.

- Core requirements (32 credits): CSE 142, 143, 321, 322, 326, 341, 370, 378
- Additional Computer Engineering requirements (17 credits): CSE 451, 461, EE 215 and 233
- Hardware Option (18 credits): CSE 467, 471, EE 331; CSE 468 or 477 (capstone)
- Software Option (16-17 credits): CSE 403, 466; one of CSE 401, 457, or 471; CSE 476 or 481 (capstone)
- CSE Electives (13 credits): all 400-level CSE courses not listed above
- Math (25 credits): MATH 124, 125, 126, 307, 308, MATH/STAT 390
- Science (20 credits): PHYS 121/131, 122/132, 123/133; CHEM 142
- VLPA and I&S (30 credits): from UW provided list.
- Written and Oral Communication (12 credits): English Composition, TC 231, 333
- Free Electives (13-15 credits; depending on option chosen)

## 5. Faculty

The department operates with minimal hierarchy. Faculty of all ranks have similar teaching loads and are similarly evaluated including 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 42 faculty, 2 of these hold partial appointments (one with EE and one with the Business School). They are broken down as follows:

- o 21 full professors
- o 5 associate professors
- o 9 assistant professors
- o 2 research assistant professors
- o 3 senior lecturers
- o 2 lecturers

All are involved with 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). Recently, one of our senior graduate student teaching assistants received a College award for teaching. 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 Bades award honors the Department's top TAs based on nominations from students.

The faculty is extremely active in research and our department is consistently ranked in the top 10 in the nation in our discipline (and usually in the top 5 or 6). 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).

Our faculty continually strives to improve its teaching performance. The Chair broadcasts anonymous histogram results of student evaluations so that all faculty members know where they stand. 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/](http://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.

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.

## 6. Facilities

### Building

CSE continues to occupy woefully inadequate space both in quality and quantity. We are housed in a 35+ year-old building with no classrooms and limited conference/meeting space. We have had to convert our hallways to breakout rooms with cubicles and whiteboards. The University administration claims that a new building with 2.5 times the assignable square feet for our department is the highest institutional priority. However, there has been no progress for several years. Given that we have converted all our available space to office or laboratory use, we are now severely limited in our ability to recruit faculty with laboratory needs and take on new initiatives. The department has traditionally assigned a high priority to undergraduate laboratory space. In fact, there are currently only 3 graduate laboratories as opposed to the 5 undergraduate laboratory spaces. The need for new facilities has been identified as our major need by both our External Advisory Committee and the University-appointed 10-year Review Committee.

### Classrooms

The classrooms used by our students and faculty are scattered throughout campus with the majority being in the new Mary Gates Hall for Undergraduate Education and the new Electrical Engineering building. Classrooms in both buildings are state-of-the-art with networking taps and data projection equipment in each room. However, the inconsistency of room assignments (several of our courses each quarter are assigned classrooms 10-15 minutes across campus) makes it difficult to rely on a standard infrastructure. Many of our faculty have not taken steps to use educational technology because of this uncertainty.

### Laboratories and Infrastructure

Laboratory and computing facilities available to computer engineering students include two general purpose modern computing laboratories, and three special purpose state-of-the art laboratories geared to specific areas in the computer engineering curriculum.

The general purpose computing labs provide approximately 75 Pentium P-II computers running Windows 2000 and a number of laser printers, and are shared by all computer science and engineering students. The PCs have 550MHz & 700MHz CPUs, 128mb RAM and 19" monitors (with anticipated upgrades to 1GHz-class machines in the next academic year). These labs support general computer science and engineering courses, as well as certain computer engineering courses that do not require special hardware, and offer a very wide array of general purpose software development tools (e.g., MS Visual Studio, C, C++, Java), as well as specialized hardware design tools (e.g., DesignWorks, L-Edit, PSpice, and Xilinx Foundation Tools.)

The three special purpose labs that support computer engineering courses include: the VLSI Design Lab, the Integrated Systems Design Lab (ISDL) and the Embedded Systems Lab.

The ISDL and the Embedded Systems Lab share a common space, and alternate use of that space to match course offerings. The ISDL is populated with 12 Pentium P-III computers (800MHz CPU, 256mb RAM, 19" monitors) augmented by state-of-the-art digital and embedded computer hardware/software development equipment. A large-format (11"x17") laser printer is available for producing high quality, easy-to-read design prints. The PC's are loaded with contemporary development software, such as Xilinx Foundation Tools, Keil SDT, ARM SDT, and Palm CodeWarrior, as well as a wide variety of general-purpose software design tools. Development kits available to students contain hardware to facilitate development on popular microcontroller and programmable logic platforms. In addition, a complete inventory of currently used electronic components is maintained.

The VLSI Design Lab consists of 18 P-III PCs and a laser printer and is dedicated to the VLSI Design class (CSE 468). The PCs have a 800MHz processor, 256mb RAM, 19" monitors, and trackballs, and run the Windows2000 operating system. Software includes a wide variety of software design tools for general program development; as well as specialized VLSI design packages such as Capture and PSpice (from OrCAD EDA), and DesignPro (Tanner Research).

A full time lab manager supports the special purpose laboratories, and is available to advise, consult and problem solve with students in the use of the test equipment, project hardware components and software tools.

All laboratories are backed by a very robust server and network infrastructure, consisting of Windows and Unix servers (for file serving, compute cycles, printing, email, and web services), interconnected with a 100mb/1gb switched network. A building-wide (and its neighborhood) 802.11b wireless network is also available for students, faculty, and staff.

Students have the opportunity to use all of these facilities in learning the use of modern engineering tools. Facilities and infrastructure are constantly updated to maintain student access to the most modern tools. Access to the general purpose laboratories is available throughout each student's career in the department, while access to the specialized labs restricted to students actively enrolled in specific computer engineering courses, or doing senior project or undergraduate research work sponsored by a faculty member. Award-winning examples of engineering projects produced by students using these facilities can be viewed at [www.cs.washington.edu/info/videos/](http://www.cs.washington.edu/info/videos/).

#### *Other Undergraduate Program Space*

Undergraduate students have a small lounge (400 sq. ft.) which contains several smaller office populated with equipment that can be used for independent projects. The space is quite inadequate and does not provide facilities for hardware-oriented projects (prototyping benches, instrumentation, etc.).

Recently (Spring 2001), through industry gifts and some overhead recapture, we have leased 3500 sq. ft. of off-campus space that is being used primarily for inter-disciplinary projects involving students from CSE and other departments on campus at both the undergraduate and graduate levels. See <http://point5.cs.washington.edu> for more information.

## **7. Institutional Support and Financial Resources**

The University of Washington operates on a biennial budget as does the entire state government. The department's annual operating budget is currently at \$1.2M and is further augmented by industry grant through our industrial affiliates program and from grants obtained directly by faculty. The department aggressively pursues corporate donations. Large donors include: Microsoft (software and cash), Intel (equipment and cash), Ford (cash), and Tektronix (equipment matching grants). Over 70 companies contribute fees to our industrial affiliates program providing much needed unrestricted funds.

The current funding level is adequate for the needs of our educational mission but relies on active faculty who are able to obtain large industry cash and equipment donations. Our state funding goes primarily to support personnel that keep our extensive infrastructure up and running. Students do not have to manage machines themselves and can rely on a highly available and robust infrastructure for all their work and communications.

## **8. Program Criteria**

The department has a strong faculty base on which it has built excellent educational programs that have earned it the University of Washington's inaugural Brotman Award for Educational Excellence. It is ranked among the top 10 in the nation in terms of both its research and educational reputation. It has five teaching award recipients among its faculty.

The curriculum for the Computer Engineering program meets all the ABET requirements for accreditation. Evidence of this has been provided above and in the attached appendices.



**9. Cooperative Education Criteria**

Not applicable.

## **10. General Advanced-Level Program**

No accreditation is sought at the graduate level.

## **Appendix I - Additional Program Information**

### **A. Tabular Data for Program**

Table A.1. Basic level Curriculum

Table A.2. Course and Section Size Summary

Table A.3. Faculty Workload Summary

Table A.4. Faculty Analysis

Table A.5. Support Expenditures

### **B. Course Syllabi**

### **C. Faculty Curriculum Vitae**

### **D. Student exit survey comments for the past two years**

**Table A.1. Basic-Level Curriculum**  
(Computer Engineering)

Year/Quarter	Course (Department, Number, Title)	Category (Credit Hours)			
		Math & Basic Sciences	Engineering Topics	General Education.	Other
			Check if Contains Design (✓)		
1 Yr/ 1 <sup>st</sup> Qtr	Math 124 Calc w/Anal Geometry	5	( )		
	Chem 142 General Chemistry	5	( )		
	English Composition		( )	5	
			( )		
1 Yr/ 2 <sup>nd</sup> Qtr	Math 125 Calc w/Anal Geometry	5	( )		
	Phys 121/131 Mechanics	5	( )		
	VLPA/I&S Elective		( )	5	
			( )		
1 Yr/ 3 <sup>rd</sup> Qtr	Math 126 Calc w/Anal Geometry	5	( )		
	Physics 122/132 Electromag	5	( )		
	VLPA/I&S Elective		( )	5	
2 Yr/ 1 <sup>st</sup> Qtr	Math 307 Differential Equations	3	( )		
	Phyics 123/133 Waves	5	( )		
	CSE 142 Computer Programming I		4 (✓)		
	VLPA/I&S Elective		( )	3	
2 Yr/ 2 <sup>nd</sup> 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 <sup>rd</sup> Qtr	Math/Stat 390 Prob & Statistics	4	( )		
	EE 215 Fundamentals of EE		4 ( )		
	TC 231 Intro. to Technical Writing		( )	3	
	VLPA/I&S Elective		( )	5	
3 Yr/ 1 <sup>st</sup> Qtr	CSE 321 Discrete Structures		4 ( )		
	CSE 370 Intro to Digital Design		4 (✓)		
	EE 233 Circuit Theory		5 (✓)		
	Free Electives		( )		3
3 Yr/ 2 <sup>nd</sup> Qtr	CSE 326 Data Structures		4 (✓)		
	CSE 378 Mach Org/Assembly Lang		4 (✓)		
	EE 331 (Hardware option) or CompE Elective (Software option)		5 (✓)		
			( )		
3 Yr/ 3 <sup>rd</sup> Qtr	CSE 341 Programming Languages		4 (✓)		
	CSE 322 Formal Models		4 ( )		
	VLPA/I&S Elective		( )	5	
	Free Elective		( )		3

(continues on next page)

**Table A.1. Basic-Level Curriculum**  
(Computer Engineering – cont'd)

Year/Quarter	Course (Department, Number, Title)	Category (Credit Hours)			
		Math & Basic Sciences	Engineering Topics	General Education.	Other
			Check if Contains Design (✓)		
4 Yr/ 1 <sup>st</sup> Qtr <b>Hardware Option</b>	CSE 451 Operating Systems		4 (✓)		
	CSE/EE 461 Networks		4 (✓)		
	CompE Elective		4 ( )		
	Free Elective		( )		3
4 Yr/ 2 <sup>nd</sup> Qtr	CSE 467 Advanced Digital Design		4 (✓)		
	CSE 471 Design & Organization		4 (✓)		
	CompE Elective		3 ( )		
	CompE Elective		3 ( )		
4 Yr/ 3 <sup>rd</sup> Qtr	CSE 468 VLSI or CSE 477 Digital Systems Design Capstone		5 (✓)		
	TC 333 Adv. Tech Writing & Oral Presentation		( )	4	
	CompE Elective		3 ( )		
	Free Elective		( )		3
4 Yr/ 1 <sup>st</sup> Qtr <b>Software Option</b>	CSE 403 Software Engineering		4 (✓)		
	CSE 451 Operating Systems		4 (✓)		
	CompE Elective		4 ( )		
	Free Elective		( )		3-4
4 Yr/ 2 <sup>nd</sup> Qtr	CSE/EE 461 Networks		4 (✓)		
	CSE 466 Software for Embedded Systems		4 ( )		
	CSE 401 Compilers or CSE 457 Graphics or CSE 471 CPU Design		3-4 (✓)		
	Free Electives		( )		4
4 Yr/ 3 <sup>rd</sup> Qtr	CSE 476 Embedded Systems or CSE 481 Software Design Capstone		5 (✓)		
	TC 333 Adv. Tech Writing & Oral Presentation		( )	4	
	CompE Elective		4 ( )		
TOTALS-ABET BASIC-LEVEL REQUIREMENTS		45	79-81 (55-56)	42	12-14
OVERALL TOTAL FOR DEGREE		180			
PERCENT OF TOTAL		25%	44-45% (31%)	23%	7-8%
MINIMUM PERCENTAGE		25%	37.5 %		

**Table A.2. Course and Section Size Summary**  
(Computer Engineering)

Course No.	Title	No. of Sections offered in Current Year	Avg. Section Enrollment	Type of Class		
				Lecture	Laboratory	Recitation Other
100	Computer Fluency	14	10	60		40
142	Computer Programming I	66	26	75		25
143	Computer Programming II	41	25	60		40
321	Discrete Structures	6	25	75		25
322	Formal Models	3	51	100		
326	Data Structures	7	22	75		25
341	Programming Languages	6	27	75		25
370	Introduction to Digital Design	6	28	75		25
373	Data Structures and Algorithms	3	65	75		25
378	Machine Org. and Assembly Lang	7	23	75		25
401	Introduction to Compiler Construction	2	40	100		
403	Software Engineering	1	53	75		25
410	Computer Systems	1	59	100		
413	Prog. Langs and Implementation	1	53	100		
415	Introduction to Artificial Intelligence	1	59	100		
417	Algorithms and Computational Complexity	1	49	100		

(continues on next page)

**Table A.2. Course and Section Size Summary**  
(Computer Engineering – cont'd)

Course No.	Title	No. of Sections offered in Current Year	Avg. Section Enrollment	Type of Class			
				Lecture	Laboratory	Recitation	Other
421	Introduction to Algorithms	1	45	100			
431	Introduction to Theory of Computation	1	38	100			
444	Introduction to Database Systems	2	52	100			
451	Introduction to Operating Systems	6	25	75		25	
457	Computer Graphics	2	52	75			25
458	Computer Animation	1	15	60			40
461	Introd. to Computer Networks (CSE only)	5	17	75		25	
466	Software for Embedded Systems	2	7	75	25		
467	Advanced Logic Design	2	17	75	25		
468	Very Large Scale Integration	1	20	60			40
471	Computer Design and Org.	2	19	75		25	
473	Introduction to Artificial Intelligence	2	45	75			25
476	Embedded System Design	2	7	60	40		
477	Digital System Design	2	15	60	40		
481	Capstone Software Design	1	25	60			40

(continues on next page)

**Table A.2. Course and Section Size Summary**

(Computer Engineering – cont'd)

Course No.	Title	No. of Sections offered in Current Year	Avg. Section Enrollment	Type of Class			
				Lecture	Laboratory	Recitation	Other
490ab	Computers, Ethics, and Society	1	8	100			
490ca	Computer Animation	1	14	60			40
490i	Advanced Internet Systems	1	30	60			40
490kt	Computer Telephony	1	25	60			40



**Table A.3. Faculty Workload Summary**  
(Computer Engineering)

Faculty Member (Name)	FT or	Classes Taught (Course No./Credit Hrs.) Term and Year	Total Activity Distribution		
			Teaching	Research	Other
Anderson, Richard	FT	A00 – 142 (4); W01 – 590Z (1-3); S01 – 521 (3)	33	50	17
Anderson, Tom	FT	A99 – 451 (4)	33	50	17
Arnstein, Larry	FT	A00 – 466 (4); W01 – 467 (4)	33	50	17
Baer, Jean-Loup	FT	A99 – 378 (4); W00 – 471 (4)	33	50	17
Beame, Paul	FT	A00 – 533 (3); W01 – 417 (3); S01 – 321 (4)	33	50	17
Borning, Alan	FT	A00 – 100 (5); S01 – 510 (3)	33	50	17
Borriello, Gaetano	FT	A00 – 370 (4); W01 – 476 (5)	33	50	17
Chambers, Craig	FT	A00 – 505 (3); W01 – 501 (3); S01 – 341 (4)	33	50	17
Curlless, Brian	FT	A00 – 457 (4); S01 – 558 (3)	33	50	17
Dickey, Martin	FT	A00 – 142 (4); 590IT (1); W01 – 143 (5); S01 – 370 (4)	80	0	20
Diorio, Chris	FT	W01 – 468/568 (5); S01 – 477 (5)	33	50	17
Domingos, Pedro	FT	A00 – 473 (3); W01 – 590PD (1-3); S01 – 592 (4)	33	50	17
Dooley, Emer	PT	A00 – 590IP (4); S01 – 590YB (4)	80	0	20
Dugan, Ben	FT	W01 – 378 (4); S01 – 142 (4)	80	0	20
Ebeling, Carl	FT	W00 – 467 (3); S00 – 477 (5)	33	50	17
Eggers, Susan	FT	A00 – 471 (4); S01 – 378 (4)	33	50	17
Eizioni, Oren	FT	A00 – 573 (3); W01 – 490I (5)	33	50	17
Fox, Dieter	FT	A00 – 590DF (1); W01 – 321 (4); S01 – 473 (3)	33	50	17

(continues on next page)

**Table A.3. Faculty Workload Summary**  
(Computer Engineering – cont'd)

Faculty Member (Name)	FT or	Classes Taught (Course No./Credit Hrs.) Term and Year	Total Activity Distribution		
			Teaching	Research	Other
Gribble, Steve	FT	W01 – 451 (4); S01 – 551 (3)	33	50	17
Halevy, Alon	FT	A00 – 594 (4); S01 – 326 (4)	33	50	17
Karlin, Anna	FT	A00 – 589 (4); W01 – 321 (4), 522 (3); S01 – 590Z (1-3)	33	50	17
Kautz, Henry	FT	A00 – 326 (4); W01 – 574 (3); S01 – 590HK (1-3)	33	50	17
Ladner, Richard	FT	A00 – 531 (3); W01 – 322 (3); S01 – 431 (3)	33	50	17
Lazowska, Ed	FT	Chair	0	0	100
Levy, Henry	FT	W00 – 451 (4)	33	50	17
Mones, Barbara	FT	W01 – 458 (4), 490AP (1-3); S01 – 490 CA (5)	80	0	20
Notkin, David	FT	A00 – 401 (3); W01- 590N (1); S01 – 584 (4)	33	50	17
Perkins, Hal	FT	A00 – 143 (5); W01 – 142 (4), 413 (3); S01 – 143 (5)	80	0	20
Popovic, Zoran	FT	W01 – 557 (3); S01 – 457 (4)	33	50	17
Rao, Rajesh	FT	W01 – 599 (4); S01 – 373 (3)	33	50	17
Ruzzo, Larry	FT	A00 – 322 (3); W01 – 421 (3); S01 – 401 (3)	33	50	17
Salesin, David	FT	at Microsoft Research	33	50	17
Seitz, Steve	FT	W01 – 590SS (3); S01 – 591 (4)	33	50	17
Sengupta, Rimli	FT	A00 – 321 (4); S01 – 322 (3)	33	50	17
Shapiro, Linda	FT	A00 – 373 (3); W01 – EE562 (3); S01 – 576 (3)	33	50	17

(continues on next page)

**Table A.3. Faculty Workload Summary**  
(Computer Engineering – cont'd)

Faculty Member (Name)	FT or	Classes Taught (Course No./Credit Hrs.) Term and Year	Total Activity Distribution		
			Teaching	Research	Other
Snyder, Larry	FT	A00 – 378 (4); W01 – 548 (3); S01 – 100 (5)	33	50	17
Suciu, Dan	FT	A00 – 444 (3); W01 – 544 (3); S01 – 590DS (3)	33	50	17
Tanimoto, Steve	FT	A00 – 341 (4); W01 – 373 (3); S01 – 415 (5)	33	50	17
Tompa, Martin	FT	A99 – 321 (4); S00 – 431 (3)	33	50	17
Weld, Daniel	FT	W00 – 490I (5)	33	50	17
Wetherall, David	FT	W01 – 461 (4)	33	50	17
Zahorjan, John	FT	A00 – 451 (4); W01 – 142 (4); S01 – 481 (5)	33	50	17

**Table A.4. Faculty Analysis**  
(Computer Engineering)

Name	Age	Rank	FT or PT	Highest Degree	Institution from which Highest Degree Earned & Year	Years of Experience			Professional Registration (Indicate State)	Level of Activity (high, med, low, none) in:		
						Govt./Industry Practice	Total Faculty	This Institution		Professional Society	Research	Consulting/Summer Work in Industry
Richard Anderson	42	Professor	FT	PhD	Stanford, 1985	0	15	15		L	M	N
Thomas Anderson	39	Professor	FT	PhD	Washington, 1991	4	10	4		M	M	L
Larry Arnstein	37	Research Ass't Professor	FT	PhD	CMU, 1993	10	2	2		N	H	M
Jean-Loup Baer	64	Professor	FT	PhD	UCLA, 1968	0	36	32		H	H	N
Paul Beame	41	Professor	FT	PhD	Toronto, 1987	0	14	14		M	H	L
Alan Borning	51	Professor	FT	PhD	Stanford, 1979	3	21	21		M	H	L
Gaetano Borriello	42	Professor	FT	PhD	Berkeley, 1988	3	13	13		L	H	M
Craig Chambers	36	Associate Professor	FT	PhD	Stanford, 1992	0	10	10		M	H	N
Brian Curless	34	Assistant Professor	FT	PhD	Stanford, 1997	0	3	3		L	H	L
Martin Dickey	53	Senior Lecturer	FT	PhD	Arizona State, 1992	14	11	5		M	L	N
Chris Diorio	39	Associate Professor	FT	PhD	Caltech, 1997	6	4	4		M	H	M
Pedro Domingos	35	Assistant Professor	FT	PhD	UC-Irvine, 1997	0	4	2		L	H	L
Emer Dooley	41	Lecturer	PT	PhD	Washington, 2000	8	1	1		L	L	N

(continues on next page)

**Table A.4. Faculty Analysis**  
(Computer Engineering – cont'd)

Name	Age	Rank	FT or PT	Highest Degree	Institution from which Highest Degree Earned & Year	Years of Experience		Professional Registration (Indicate State)	Level of Activity (high, med, low, none) in:		
						Govt./Industry Practice	Total Faculty		This Institution	Professional Society	Research
Ben Dugan	31	Lecturer	FT	MS	Washington, 1995	2	3		N	L	M
Carl Ebeling	51	Professor	FT	PhD	CMU, 1986	0	15		M	H	L
Susan Eggers	58	Professor	FT	PhD	Berkeley, 1989	4	12		L	H	L
Oren Eizoni	37	Associate Professor	FT	PhD	CMU, 1990	0	10		M	M	M
Dieter Fox	34	Assistant Professor	FT	PhD	Bonn, 1998	0	3		L	H	L
Steve Gribble	28	Assistant Professor	FT	PhD	Berkeley, 2000	0	1		L	H	N
Alon Halevy	37	Associate Professor	FT	PhD	Stanford, 1993	4	4		H	H	H
Anna Karlin	41	Professor	FT	PhD	Stanford, 1987	6	7		M	H	N
Henry Kautz	44	Associate Professor	FT	PhD	U-Rochester, 1988	13	1.33		H	H	H
Richard Ladner	57	Professor	FT	PhD	Berkeley, 1971	0	30		L	H	M
Edward Lazowska	50	Professor	FT	PhD	Toronto, 1977	0	24		H	L	H
Henry M. Levy	48	Professor	FT	MS	Washington, 1981	8	18		H	H	L
Barbara Mones-Haftal	49	Senior Lecturer	FT	MFA	Rhode Island School of Design, 1979	3	15		H	L	M

(continues on next page)

**Table A.4. Faculty Analysis**  
(Computer Engineering – cont'd)

Name	Age	Rank	FT or PT	Highest Degree	Institution from which Highest Degree Earned & Year	Years of Experience			Professional Registration (Indicate State)	Level of Activity (high, med, low, none) in:		
						Govt./Industry Practice	Total Faculty	This Institution		Professional Society	Research	Consulting/Summer Work in Industry
David Notkin	46	Professor	FT	PhD	CMU, 1984	0	17	17		H	H	M
John Hal Perkins	49	Senior Lecturer	FT	MS	Cornell, 1982	2	18	3		L	N	N
Zoran Popovic	32	Assistant Professor	FT	PhD	CMU, 1999	0	2	2		N	H	N
Rajesh Rao	31	Assistant Professor	FT	PhD	Rochester, 1998	2	1	1		M	H	L
Walter L. Ruzzo	54	Professor	FT	PhD	Berkeley, 1978	0	24	24		N	H	N
David Salesin	40	Professor	PT	PhD	Stanford, 1991	1	10	10		L	H	H
Steven Seitz	32	Assistant Professor	FT	PhD	Wisconsin, 1997	1	4	1		L	H	L
Rimli Sengupta	35	Research Ass't Prof.	FT	PhD	Georgia Tech, 1995	0	7	2		L	H	N
Linda Shapiro	51	Professor	FT	PhD	Iowa, 1974	0	27	15		M	H	L
Lawrence Snyder	54	Professor	FT	PhD	CMU, 1973	0	28	18		M	H	L
Dan Suciu	42	Assistant Professor	FT	PhD	Pennsylvania, 1995	5	10	1		L	H	M
Steve Tanimoto	51	Professor	FT	PhD	Princeton, 1975	0	26	24		M	H	H
Martin Tompa	49	Professor	FT	PhD	Toronto, 1978	0	23	23		L	H	H

(continues on next page)

**Table A.4. Faculty Analysis**  
(Computer Engineering – cont'd)

Name	Age	Rank	FT or PT	Highest Degree	Institution from which Highest Degree Earned & Year	Years of Experience			Professional Registration (Indicate State)	Level of Activity (high, med, low, none) in:		
						Govt./Industry Practice	Total Faculty	This Institution		Professional Society	Research	Consulting/Summer Work in Industry
Dan Weld	40	Professor	FT	PhD	MIT, 1988	0	13	13		H	H	H
David Wetherall	34	Assistant Professor	FT	PhD	MIT, 1998	0	2	2		N	H	L
John Zahorjan	46	Professor	FT	PhD	Toronto, 1980	0	21	21		L	M	L

Note that above lists all faculty in the department. It is not possible to separate faculty into Computer Science program and Computer Engineering program faculty. Approximately 50% of our current undergraduates are expected to graduate with a Computer Engineering degree.

**Table A.5. Support Expenditures**  
(Computer Engineering)

Fiscal year	1	2	3	4
	<b>1998-99</b> (prior to previous year)	<b>1999-00</b> (previous year)	<b>2000-01</b> current year)	<b>2001-02</b> (year of visit)
<b>Expenditure Category</b>				
Operations (1) (not including staff)	675,384	1,114,148	1,206,383	1,254,638
Travel (2)	78,204	107,877	100,136	104,141
Equipment (3)	130,921	788,023	1,018,620	1,059,365
(a) Institutional Funds	30,844	590,486	373,943	388,901
(b) Grants and Gifts (4)	100,077	197,537	644,678	670,465
Graduate Teaching Assistants	365,275	518,549	385,003	400,403
Part-time Assistance (5) (other than teaching)	238,209	337,743	231,798	241,070
<b>Total</b>	<b>1,487,993</b>	<b>2,866,340</b>	<b>2,941,940</b>	<b>3,059,618</b>

**All amounts are for the entire department. Computer Engineering students account for approximately 50% of our students in 2001-2 and 33% in 1998-1999 with linear scaling in between.**

Comments:

Graduate Teaching Assistants and Part-time Assistance excludes fringe benefits.  
Above amounts include Research Support allocations.

Sources:

Funding types include General Operating Funds and Designated operating funds, 74- and 75-;  
Actual exp operations, travel, equipment institutional funds, GTAs, part-time; Fray Database;  
Equipment (b): Office of Institutional Studies financial database;  
2000-01 includes actuals thru month 8 and a 33% factor for the remainder of the year;  
2001-02 data is projected @ 104% of 2000-01.



# Appendix I

## B. Course Syllabi

### Core courses:

- CSE 142: Computer Programming for Engineers and Scientists I
- CSE 143: Computer Programming for Engineers and Scientists II
- CSE 321: Discrete Structures
- CSE 322: Introduction to Formal Models in Computer Science
- CSE 326: Data Structures
- CSE 341: Programming Languages
- CSE 370: Introduction to Digital Design
- CSE 378: Machine Organization and Assembly Language Programming

### Senior courses:

- CSE 401: Introduction to Compiler Construction
- CSE 403: Software Engineering
- CSE 421: Introduction to Algorithms
- CSE 431: Introduction to Theory of Computation
- CSE 444: Introduction to Database Systems
- CSE 451: Introduction to Operating Systems
- CSE 457: Computer Graphics
- CSE 458: Computer Animation
- CSE 461: Introduction to Computer-Communication Networks
- CSE 466: Software for Embedded Systems
- CSE 467: Advanced Digital Design
- CSE 471: Computer Design and Organization
- CSE 473: Introduction to Artificial Intelligence
- CSE 490ab: Computers, Ethics, and Society
- CSE 490ca: Computer Animation
- CSE 490i: Advanced Internet Systems
- CSE 490kt: Computer Telephony

### Capstone design courses:

- CSE 468: Very Large Scale Integration
- CSE 476: Embedded System Design
- CSE 477: Digital System Design
- CSE 481: Capstone Software Design

### Independent study and/or research:

- CSE 498: Senior Project (Honors: 498H)
- CSE 499: Reading and Research

### Non-majors courses (included only for completeness):

- CSE 100: Fluency in Information Technology
- CSE 373: Data Structures and Algorithms
- CSE 410: Computer Systems
- CSE 413: Programming Languages and Their Implementation
- CSE 415: Introduction to Artificial Intelligence
- CSE 417: Algorithms and Computational Complexity



### Core courses

- CSE 142: Computer Programming for Engineers and Scientists I
- CSE 143: Computer Programming for Engineers and Scientists II
- CSE 321: Discrete Structures
- CSE 322: Introduction to Formal Models in Computer Science
- CSE 326: Data Structures
- CSE 341: Programming Languages
- CSE 370: Introduction to Digital Design
- CSE 378: Machine Organization and Assembly Language Programming

# CSE 142: Computer Programming for Engineers and Scientists I (4)

---

## Catalog description:

Basic programming-in-the-small abilities and concepts. Highlights include procedural and functional abstraction with simple built-in data type manipulation. Basic abilities of writing, executing, and debugging programs. Not available for credit to students who have completed CSE 210 or ENGR 141.

**Prerequisites:** none

## Textbook(s) and/or other required material:

Jeri Hanly & Elliot Koffman, "Problem Solving & Program Design in C", Addison-Wesley, 3rd ed, 1999.

## Course objectives:

Develop basic programming and problem solving skills in a procedural language

## Topics covered:

- Basic concepts of computer structure and program execution
- Variables, types, expressions, and assignment
- Input/output: console and monitor, text files
- Conditional execution (if, switch)
- Iteration (while, for)
- Loop development and design patterns
- Functions, including value and reference (pointer) parameters
- Program organization and libraries
- Use of standard C and local libraries
- Analyzing and depicting function relationships
- Functional decomposition of problems
- Arrays (1- and 2-D)
- Sequential and binary search
- Sorting (insertion or selection sort)
- Structures
- User-defined types combining arrays and structs
- Strings
- Introduction to recursion
- Graphics concepts
- Concepts of event-driven programming
- Programming style
- Experience with team or group work

## Class/laboratory schedule:

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

---

# CSE 143: Computer Programming for Engineers and Scientists II (5)

---

## Catalog description:

Continuation of 142. Concepts of modularity and encapsulation, focusing on modules and abstract data types. Covers some basic data structures. Not available for credit to students who have completed CSE 211.

**Prerequisites:** none

## Textbook(s) and/or other required material:

Frank Carrano, Paul Helman, and Robert Veroff, "Data Abstraction and Problem Solving with C++: Walls and Mirrors," Addison-Wesley, 2nd ed., 1998.

## Course objectives:

Develop intermediate programming and problem-solving skills, including introduction to object-oriented programming; fundamental data structures and algorithms; and experience working in teams with programs of considerable complexity.

## Topics covered:

- Introduction to C++ (in contrast of C)
- Abstract data types
- Dynamic storage allocation and pointers
- C++ classes: as a technique for encapsulation and abstraction
- C++ classes: technical apparatus, including constructors, destructors, assignment, etc.
- Classes and objects, inheritance, and dynamic dispatch
- Basics of object-oriented design
- Recursion
- Linked lists
- Basic container types: lists, stacks, and queues
- Program efficiency, big-O notation
- Searching and sorting (quicksort, mergesort)
- Trees, particularly binary and binary search trees
- Survey of hashing and other container types
- Modeling and simulation concepts
- Principles of engineering robust and extensible software (documentation, testing, modularity, reusability, etc.)
- Experience working in a team or group

## Class/laboratory schedule:

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

---

# CSE 321: 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:

Rosen, Discrete Mathematics and Its Applications, Fourth 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

## Class/laboratory schedule:

Meetings: 3 1-hour lectures per week, 1 recitation section hour per week  
Assignments: weekly written assignments  
Midterm plus final

---

# CSE 322: 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:

Either: Lewis & Papadimitriou Elements of the Theory of Computation: Second Edition, Prentice Hall, 1998 or: Michael Sipser, Introduction to the Theory of Computation, PWS Publishing, 1997.

## 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.

## Topics covered:

strings and languages, concatenation, powers, and reversal of strings  
operations on languages: union, concatenation and star  
regular languages, regular expressions  
countability of the regular languages, countability and uncountability, (Cantor's proof)  
context-free grammars  
examples from real programming languages such as C++  
derivation trees, leftmost derivations  
ambiguous grammars, inherently ambiguous context-free grammars  
non-context-free languages w/o proof  
closure Properties - union, concatenation, star  
decision problems - emptiness, finiteness, equivalence  
parsing methods, top-down/bottom-up parsing, deterministic/non-deterministic parsing  
finite automata, DFAs, NFAs  
subset construction  
Kleene's theorem, closure properties - emptiness, finiteness, equivalence  
decision problems  
cross product construction  
nonregular languages

## Class/laboratory schedule:

3 lecture hours per week. Weekly written assignments. Tests.

---

# CSE 326: 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 EE 374.

**Prerequisites:** CSE 321.

## Textbook(s) and/or other required material:

Data Structures and Algorithm Analysis in C++, Mark A. Weiss,  
Benjamin Cummings Publishing Co., 1999

## Course objectives:

The objective of this class is to study the fundamental data structures and algorithms used in computer science.

## Topics covered:

1. Introduction: Data Structures, Abstraction, Induction, Recursion
2. Algorithm Analysis
3. Lists, Stacks, and Queues
4. Trees, Search Trees: Binary, AVL, Splay, B-tree
5. Hashing
6. Priority Queues: Heaps
7. Sorting: Mergesort, Quicksort, Decision
8. Disjoint Sets: Union-Find
9. Graph Algorithms

## Class/laboratory schedule:

3 hours per week of lecture  
1 hour per week of discussion section

---



# CSE 341: 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 of:

Abelson and Sussman, Structure and Interpretation of Computer Programs

Simon Thompson, Miranda: The Craft of Functional Programming

Jeff Ullman, Elements of ML Programming

And one of:

Timothy Budd, Understanding Object-Oriented Programming with Java

Mark Guzdial, Squeak: Object-Oriented Programming with Multimedia Applications

And sometimes:

Kim Marriott and Peter J. Stuckey, Programming with constraints : an introduction

## Course objectives:

To learn fundamental programming language concepts and to gain practical experience with a diverse set of programming languages.

## Topics covered:

Programming language design principles (readability, orthogonality, regularity, etc.)

Interpreters, browsers

Recursive functions over dynamic, recursive data structures (e.g. lists, trees)

Functional programming: side-effect-free programming; programming with first-class functions; lexical scoping of nested functions

Object-oriented programming: classes, objects, methods, instance variables; inheritance, overriding, overloading; object-oriented design principles

Type checking: static vs. dynamic typing; polymorphic static type systems; type inference; subtyping

## Class/laboratory schedule:

3 lectures per week

1 section per week

---

# CSE 370: 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:

Randy H. Katz, Contemporary Logic Design, Addison/Wesley, 1994

## 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: 2 lectures

Combinational Logic: 5 lectures

Combinational Logic Implementation: 4 lectures

Combinational Logic Examples: 4 lectures

Hardware Description Languages: 2 lectures

Sequential Logic and Simple Registers: 4 lectures

Sequential Logic Implementation and Finite State Machines: 4 lectures

Sequential Logic Examples: 3 lectures

Computer Organization: 2 lectures

## Class/laboratory schedule:

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

---

# **CSE 378: Machine Organization and Assembly Language Programming (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.

## **Textbook(s) and/or other required material:**

D. Patterson and J. Hennessy:  
Computer Organization & Design: The Hardware/Software Interface  
(2nd Edition, 1998)

## **Course objectives:**

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

## **Topics covered:**

machine organization, instruction sets, addressing modes, instruction encoding, subroutine handling, pipelining, microprogramming, memory systems, caches, I/O

## **Class/laboratory schedule:**

3 lectures/week  
1 discussion section/week

---

## Senior courses

- CSE 401: Introduction to Compiler Construction
- CSE 403: Software Engineering
- CSE 421: Introduction to Algorithms
- CSE 431: Introduction to Theory of Computation
- CSE 444: Introduction to Database Systems
- CSE 451: Introduction to Operating Systems
- CSE 457: Computer Graphics
- CSE 458: Computer Animation
- CSE 461: Introduction to Computer-Communication Networks
- CSE 466: Software for Embedded Systems
- CSE 467: Advanced Digital Design
- CSE 471: Computer Design and Organization
- CSE 473: Introduction to Artificial Intelligence
- CSE 490ab: Computers, Ethics, and Society
- CSE 490ca: Computer Animation
- CSE 490i: Advanced Internet Systems
- CSE 490kt: Computer Telephony

# CSE 401: 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:

"Compilers: Principles, Techniques, and Tools" by Aho, Sethi, and Ullman

## 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 C++.

## Topics covered:

Organization of Compilers and Interpreters  
Lexical Analysis  
Syntactic Analysis  
Semantic Analysis  
Interpretation  
Run-Time Storage Layout  
Code Generation  
Optimization

## Class/laboratory schedule:

Lecture, 3 hours per week.  
Laboratory, as needed to complete course project.

---

# CSE 403: 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 321; CSE 341; CSE 378; recommended: CSE 401; CSE 451.

## **Textbook(s) and/or other required material:**

FUNDAMENTALS OF SOFTWARE ENGINEERING  
Carlo GHEZZI, Mehdi JAZAYERI, Dino MANDRIOLI  
Publisher : Prentice Hall - Englewood Cliffs, NJ  
ISBN: 0-13-820432-2

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.

## **Class/laboratory schedule:**

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 guest lectures from local industry during the last week or so of the quarter (when the students are completely focused on their project). The sections are used for reviews of the projects and for presentation by the TA of material on specific tools and technologies needed to develop the project.

---

# CSE 421: 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:

Varies. Cormen, Leiserson, Rivest "Intro to Algorithms" or Manber, same title, are commonly used.

## Course objectives:

Learn basic techniques for design and analysis of algorithms, including correctness proofs. Learn a number of important basic algorithms. NP-complete and other intractable problems.

## Topics covered:

Main Techniques:

Design: Induction, Graph search, Divide and Conquer, Greedy, Dynamic Programming, Branch and Bound.

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

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

## Class/laboratory schedule:

3 50 minute lectures per week.

---

# CSE 431: 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", PWS Publishing, 1997.

## 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  
L vs. NL, NL-completeness, Savitch's Theorem, Immerman/Szelepcseny Theorem

## Class/laboratory schedule:

Three 50-minute lectures per week.

---



# CSE 444: 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:**

Database Management Systems (Ramakrishnan and Gehrke)

A first course on database systems (Ullman, Widom and Garcia Molina)

## **Course objectives:**

Provide a basic introduction to the concepts of database system design and implementation, including: data models, conceptual design, query languages, system components, data storage, query optimization and transaction processing.

## **Topics covered:**

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

## **Class/laboratory schedule:**

3 lectures a week

---

# CSE 451: 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 EE 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 large operating systems project (using Nachos).

## **Topics covered:**

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

## **Class/laboratory schedule:**

Class: 3 days per week, 1 hour  
Sections: 1 day per week, 1 hour  
Labs: as needed for project and assignments

---

# CSE 457: Computer Graphics (4)

---

## **Catalog description:**

Introduction to computer image synthesis and interactive computer graphics applications. Topics include computer graphics hardware, color image display, event-driven programming, line drawing, polygon scan conversion, texture mapping, image morphing, image compositing, curves and surfaces, hidden surface algorithms, local illumination models, ray tracing, and photorealistic image synthesis.

**Prerequisites:** CSE 326.

## **Textbook(s) and/or other required material:**

Alan Watt, 3D Computer Graphics, Third Edition, Addison-Wesley, 2000.

## **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 homeworks and projects.

## **Topics covered:**

Topics include computer graphics hardware, color image display, event-driven programming, line drawing, polygon scan conversion, texture mapping, image morphing, image compositing, curves and surfaces, hidden surface algorithms, local illumination models, ray tracing, and photorealistic image synthesis.

## **Class/laboratory schedule:**

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

---

# CSE 458: 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 or 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

## **Class/laboratory schedule:**

Tuesday and Thursday 10:30 - 11:20 Lecture  
Lab Hours in LA2

---

# CSE 461: Introduction to Computer-Communication Networks (4)

---

## Catalog description:

Computer network architectures, protocol layers. Transmission media, encoding systems, error detection, multiplexing, switching. Data link, multiple access channel protocols. Methods for network routing, congestion control, flow control. End-to-end transport services, protocols. Network security, privacy. Applications including electronic mail, virtual terminals, distributed operating systems. Network programming.

**Prerequisites:** CSE 143.

## Textbook(s) and/or other required material:

"Computer Networks: A Systems Approach", 2nd 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 internetworking ground in-between these two extremes. We will cover 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), quality of service, multicast, and security.

## Class/laboratory schedule:

Two 1 1/2 hour classes per week, laboratory available for projects.

---

# CSE 466: 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:

In inaugural course, we used An Embedded Software Primer, David E. Simon, Addison Wesley. But, will likely not require this book in the future. Extensive use of technical datasheets and on-line documentation of devices and tools.

## Course objectives:

Hand's on experience in the design, development, and analysis of embedded software systems. Including basics of embedded hardware

## Topics covered:

- Embedded Hardware
- Peripheral Interfaces (Embedded Hardware, I/O, power, clocks, etc.)
- The 8051 Embedded Processor, and Development Tools
- Timers and Interrupts
- Embedded Software Architectures and Multi-tasking
- Embedded Operating Systems
- Device Drivers (Linux style)
- PDA Programming and Interfacing
- Timing analysis
- Safety, and Society
- Communications and Embedded Networking

## Class/laboratory schedule:

- 3 hours lecture
- 3 hours lab

---

# CSE 467: 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:

Contemporary Logic Design, R. H. Katz, Benjamin-Cummings/Addison-Wesley 1994.

## 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:

1. Review of basic digital-logic design; Combinational logic; Structured logic implementation; Sequential logic; Finite-state machines
2. Overview of digital technology; Logic families; Reading and understanding data books; Interfacing; Fixed-function devices; TTL/CMOS; glue logic; RAM/ROM; Programmable devices; PROMs; PALs and PLDs; FPGAs; Integrated circuits
3. Electrical realities; Resistance, capacitance and inductance; Time constants; Decoupling and ground; Power dissipation and drops; Wire delays; Fanout and loading; Ringing, reflections, and terminations; Clock
4. Computer-aided design; Hardware description languages; Logic compilation; Two-level and multi-level logic synthesis; Technology-independent optimization; Technology mapping; Sequential-logic synthesis; Tools for mapping to PLDs and FPGAs;
5. Laboratory realities; Logic analyzer and oscilloscope basics; Repetitive versus single-shot triggering; Timing, state, capture, bandwidth; Glitches and transient events
6. System-level components; Static, dynamic, and nonvolatile memories; Memory controllers and timing; Digital communication; Serial and parallel protocols; Synchronous vs. asynchronous data communication; Busses; Arbitration schemes
7. Technology; MOSFETs; FPGAs; Integrated circuits; Circuit boards; High-speed circuits; controlling impedances

## Class/laboratory schedule:

3 hours lecture and 3 hours lab per week

---

# CSE 471: 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", 2nd ed. Morgan-Kaufman, 1998

## 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

## Class/laboratory schedule:

3 lectures a week (50 mns)

---



# CSE 490ab: Computers, Ethics, and Society (1)

---

**Catalog description:**

none (special topics course not yet listed)

**Prerequisites:** none**Course objectives:**

To expose students to a wide range of topics in computers, ethics, and societal impacts, and to facilitate discussion, debate, and critical thinking on these topics.

**Topics covered:**

- professional ethics
- impacts of computing on the economy
- computers and education
- privacy and databases
- universal and differential access
- impacts on social capital
- free speech online
- the impact of the internet on the political process
- computer crime; hacking; attacks
- software patents and copyrights
- commercialization within a university environment
- antitrust and monopoly law and the software industry

**Class/laboratory schedule:**

1 hour seminar, once a week, often with an outside speaker

---

# CSE 490ca: Computer Animation (5)

---

**Catalog description:**

none (special topics course not yet listed)

**Prerequisites:** none

**Textbook(s) and/or other required material:**

None

**Course objectives:**

To learn the production pipeline for short animated films  
To learn how to design and implement a storyreel and animatic for a short film  
To learn how to produce a brief professionally  
produced collaborative work with the emphasis on applying the knowledge  
gained to create a unique product.  
To work with other students on a team.  
The design and implement a final project that  
incorporates and displays both aesthetic and technical excellence.

**Topics covered:**

Production Planning for Animated Short Films  
Story Art  
Storyboard and Layout Techniques  
Character Modeling  
Character Setup  
Character Animation  
Acting and Performance for Animators  
Designing shapes for Facial Expression  
Designing and Building Surface Shaders  
Compositing and Post Production  
Animation and Technical Direction  
Digital Special Effects

**Class/laboratory schedule:**

Tuesday and Thursday 10 - 11:50 Lecture  
Lab in LA2

---

# CSE 490i: Advanced Internet Systems (5)

---

## Catalog description:

none (special topics course not yet listed)

HTTP, XML, ICE and W3C standards. Design of web spiders, exploration, indexing and scalable search engine query processing. Information retrieval, TF/IDF, latent semantic indexing, hypertext link analysis, result clustering. Website management, information integration, ecommerce, personalization, collaborative filtering, security and privacy. Case studies: Alta vista, Google, Ask Jeeves, Jango, Vignette, Strudel, Amazon, Webwatcher.

**Prerequisites:** none

## Textbook(s) and/or other required material:

None

## Course objectives:

Introduction to Internet technologies  
Design and implementation of a substantial application in a team environment.

## Topics covered:

Introduction  
Networking Fundamentals  
HTTP and Server Basics  
Spiders and Search Engine Architecture  
Learning and Classification Algorithms  
Search Engines and Information Retrieval  
Clustering  
Basics of Databases  
Database Implementation  
Information Aggregation and Its Applications  
MetaCrawler  
XML  
Consistency and Availability for Large-scale Applications  
E-Commerce  
Cryptographic Primitives  
Distributed Application Network: Where the Internet and Distributed Computing Converge

---

# CSE 490kt: Computer Telephony (5)

---

**Catalog description:**

none (special topics course not yet listed)

**Prerequisites:** none**Textbook(s) and/or other required material:**

None

**Course objectives:**

Introduction to MS Automation via Computer Telephony. ActiveX Components illustrate the use of databases in CT, audio recordings, internet transmission of sounds and images, voice recognition and text-to-speech, high-level CT protocols, low-level CT controls including call-analysis and routing via SC-bus.

**Topics covered:**

MS Automation; ADO database connections; audio recordings and playback; FSM implementation of home automation; WinSock internet connections; DirectX recording, mixing and playback; voice recognition and text-to-speech; flowchart control of telephony; low-level control of telephony. A substantial course project combining 2 or more of the above topics.

---

Capstone design courses:

- CSE 468: Very Large Scale Integration
- CSE 476: Embedded System Design
- CSE 477: Digital System Design
- CSE 481: Capstone Software Design

# CSE 468: 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, J. M. Rabaey, Prentice-Hall Electronics and VLSI Series, 1996.

## Course objectives:

1. To learn 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.
2. To learn 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.
3. To learn the tools of custom IC design, including schematic entry, simulation (analog, static timing, dynamic timing), layout, DRC, and LVS. Basics of IC fabrication.

## Topics covered:

1. Introduction and overview of modern IC design
2. Transistor basics; Fermi levels and band diagrams; pn-junction diode; MOS transistor Sub- and above-threshold MOSFET operation; Static and dynamic MOSFET behavior
3. CAD tools for physical IC design; OrCAD Capture for schematic entry; Spice for simulation and analysis; Tanner Tools LEdit for physical layout; Tanner Tools DRC and LVS to verify design; UW Parasite to back-annotate simulation
4. Inverters and pass transistors; Static and dynamic behavior; Noise margin, fan-in and fan-out; Time constants, logic levels, drive
5. Static CMOS logic; Basic logic gates; Static and dynamic behavior; Time constants (RC), loading, fan-in and fan-out, delays; Transistor sizing
6. Dynamic CMOS logic; Domino, Zipper, CVSL logic; Dynamic & cascading behavior
7. Sequential circuits; Static latches and flip-flops; Dynamic latches and flip-flops; Registers; Merged logic; Clocking (1 and 2-phase); Clock skew, delays, race conditions
8. Memories; Static RAM; Sense amplifiers; Address decoders; Dynamic RAM; Nonvolatile memories
9. Arithmetic circuits; Adders; Multipliers
10. Chip design; Floorplanning; Power bussing, power and ground bounce; Interconnect and delays; Clocking and PLLs; Yield; Design margin; Pads
11. Advanced topics; Technology scaling; Low-power design; Bipolar transistors, differential CML; High-speed circuits, controlled impedance lines

## Class/laboratory schedule:

3 lectures and 3 hour laboratory per week. Additional laboratory time for projects.

---

# CSE 476: Embedded System Design (5)

---

## Catalog description:

System building course to provide students with a complete experience in embedded system design. Students will design, simulate, construct, debug, and document a substantial project of their choosing. Lectures will focus on case studies and emerging components and platforms .

**Prerequisites:** CSE 451; CSE 466.

## 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 software issues in embedded system code.

Familiarity with basic 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

Standards: Physical layer; Protocol; Packaging; Standardization process; evolution

Case Studies; Past capstone design projects; Industry case studies

## Class/laboratory schedule:

3 lectures and 3 hour laboratory per week. Lecture time for the first half of the quarter is spent covering the syllabus topics. In the last half of the quarter, lecture time is spent on design reviews, project presentations, special topics and guest lecturers. Students spend most of their out-of-class time in the lab designing, constructing and debugging their projects.

---

# CSE 477: Digital System Design (5)

---

## Catalog description:

Students use laboratory to design, simulate, construct, and debug a substantial project that includes hardware, software, and communication components. Lectures focus on use of embedded processors in digital system design and interfacing techniques. Writing and debugging of real-time reactive software emphasized.

**Prerequisites:** CSE 378; 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 hardware, software, and communication components.

Understanding of basic microcontrollers and their use in embedded system design.

Appreciation for the software issues in embedded system code.

Familiarity with basic serial and parallel communication methods.

Experience the design and development of a complete product using hardware and software, from design to implementation and debugging.

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

Ability to work toward a common goal in a team environment.

## Topics covered:

Introduction to embedded systems, Microprocessors and microcontrollers

Review of basic computer organization: Address/data bus; Memories; I/O ports

Timing subsystems

Interfacing techniques: Basic I/O ports; Interactions involving time

Polling, Interrupts and interrupt handling

Communication: Serial; Parallel; Basic wireless schemes; Error correction; Flow control

Hardware Design; Partitioning Software and Hardware

Interface design and implementation, Interface support devices

Design experiences: Case studies; Industry perspectives

Research directions

## Class/laboratory schedule:

3 lectures and 3 hour laboratory per week. Lecture time for the first half of the quarter is spent covering the syllabus topics. In the last half of the quarter, lecture time is spent on design reviews, project presentations, special topics and guest lecturers. Students spend most of their out-of-class time in the lab designing, constructing and debugging their projects.

---



# CSE 481: Capstone Software Design (5)

---

## Catalog description:

Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product.

**Prerequisites:** CSE 326; CSE 341; CSE 378 and substantial programming experience, such as in CSE 451 or 457.

## 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 modelling 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.

## Class/laboratory schedule:

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.

---

### Independent study and/or research

- CSE 498: Senior Project (Honors: 498H)
- CSE 499: Reading and Research

## **CSE 498: Senior Project (1-6)**

### **CSE 498H Honors (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:** none

---

## **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 advisor and CSE faculty member. Free elective, but does not replace core course or computer science elective. Credit/no credit only.

**Prerequisites:** none

---

Non-majors courses  
(included only for completeness)

- CSE 100: Fluency in Information Technology
- CSE 373: Data Structures and Algorithms
- CSE 410: Computer Systems
- CSE 413: Programming Languages and Their Implementation
- CSE 415: Introduction to Artificial Intelligence
- CSE 417: Algorithms and Computational Complexity



# **CSE 100: Fluency in Information Technology (5)**

---

**Catalog description:**

Introduces the skills, concepts, and capabilities necessary to effectively use information technology. Includes logical reasoning, managing complexity through the operation of computers and networks, and contemporary applications such as email and word processing. Not available for credit to students who have completed CSE 142 or ENGR 142.

**Prerequisites:** none

**Textbook(s) and/or other required material:**

Fluency With Information Technology (Draft) by Snyder  
Visual Basic 6, Eliason and Malarkey, Que, 1999

**Course objectives:**

To enable students to be "Fluent with Information Technology" as defined by the National Research Council's report, "Being Fluent with IT".

This is the first course of its kind.

**Topics covered:**

The NRC Report itemizes 10 skills, 10 concepts (including algorithmic thinking and programming) and 10 capabilities that would constitute FITness. This class covers 9 skills, 10 concepts and 8 capabilities.

**Class/laboratory schedule:**

MWF 1 hour lecture  
TTh 1 hour Closed Lab

---

# CSE 373: Data Structures and Algorithms (3)

---

## Catalog description:

Fundamental algorithms and data structures for implementation. Techniques for solving problems by programming. Linked lists, stacks, queues, directed graphs. Trees: representations, traversals. Searching (hashing, binary search trees, multiway trees). Garbage collection, memory management. Internal and external sorting. No credit to students who have completed 326, 374, or E E 374.

**Prerequisites:** CSE 143.

## Textbook(s) and/or other required material:

Data Structures and Algorithm Analysis by Mark Allen Weiss

## Course objectives:

This class covers a fairly broad sampling of fundamental data structures and algorithms used in everyday computer programming. The strengths, weaknesses, and practical uses of each new concept will be examined. The intent of the class is to equip students with a solid set of tools and skills necessary for choosing or designing appropriate data structures and algorithms used in programming.

## Topics covered:

- Algorithm Analysis
- Linked Lists
- Stacks & Queues
- Trees
- Balanced Trees
- Hashing
- Heaps
- Sorting Algorithms
- Graphs
- Search Algorithms
- Minimum Spanning Trees
- Shortest Paths
- Algorithm Classification

## Class/laboratory schedule:

MWF 1 hour lecture

---



# CSE 410: Computer Systems (3)

---

## Catalog description:

Structure and components of hardware and software systems. Machine organization, including central processor and input-output architectures; assembly language programming; operating systems, including process, storage, and file management. No credit to students who have completed 378 or 451.

**Prerequisites:** CSE 373.

## Textbook(s) and/or other required material:

Computer Organization and Design: The Hardware/Software Interface, Second Edition, by David Patterson and John Hennessy.

Operating Systems: Design And Implementation, Second Edition by Andrew S. Tanenbaum and Albert S. Woodhull.

## Course objectives:

To learn the concepts of computer architecture and operating systems. The first half of the course focuses on architecture, using the MIPS R2000 as a specific example. We study MIPS assembly language, implementation schemes, and memory hierarchy. The second half of the course focuses on operating systems, covering process, memory, and file management techniques.

## Topics covered:

- Data representations
- Instruction sets and assembly language
- Pipelining
- Memory hierarchies and caches
- Operating systems overview
- Processes and threads
- Synchronization and communications
- Deadlock
- Scheduling
- Memory management and virtual memories
- File systems

## Class/laboratory schedule:

3 hours of lectures per week.

---

# CSE 413: Programming Languages and Their Implementation (3)

---

**Catalog description:**

none

**Prerequisites:** none

**Textbook(s) and/or other required material:**

Revised Report on the Algorithmic Language Scheme (R5RS)  
Timothy Budd, "Understanding Object-Oriented Programming with Java," Addison-Wesley, updated edition 2000  
Course handouts and lecture notes

**Course objectives:**

Expose students to different ways of thinking about computation and programming, and gain insight into how programming languages are implemented.

**Topics covered:**

Functional programming using Scheme  
Object-Oriented programming using Java  
Implementation of programming languages, including introduction to formal grammars and parsing, introduction to machine organization, and implementation of a recursive-descent compiler for a small subset of C, generating x86 assembly language code (term project)

**Class/laboratory schedule:**

3 50-minute lectures per week

---

# CSE 415: Introduction to Artificial Intelligence (5)

---

## **Catalog description:**

Principles and programming techniques of artificial intelligence: LISP, symbol manipulation, knowledge representation, logical and probabilistic reasoning, learning, language understanding, vision, expert systems, and social issues. Not open for credit to students who have completed 473.

**Prerequisites:** CSE 373.

## **Textbook(s) and/or other required material:**

S. Tanimoto, The Elements of Artificial Intelligence Using Common Lisp, 2d ed. W. H. Freeman, 1995.

## **Course objectives:**

1. Survey fundamental techniques of artificial intelligence including knowledge representation, reasoning, and computer understanding of sensory and language information.
2. Provide hands-on experience designing and implementing programs that embody techniques of artificial intelligence.

## **Topics covered:**

Definitions of intelligence, the Turing test, LISP, symbol manipulation, techniques for web-based intelligent agents, knowledge representation, logical and probabilistic reasoning, commonsense reasoning methods, learning, natural language understanding, computer vision, expert systems, and social issues.

## **Class/laboratory schedule:**

MWF 12:30-1:20. Most classes are lecture format, but they occasionally include in-class exercises, quizzes, and 2-3 class meetings are held in a lab.

---

# CSE 417: Algorithms and Computational Complexity (3)

---

## Catalog description:

Design and analysis of algorithms and data structures. Efficient algorithms for manipulating graphs and strings. Fast Fourier Transform. Models of computation, including Turing machines. Time and space complexity. NP-complete problems and undecidable problems

**Prerequisites:** CSE 373.

## Textbook(s) and/or other required material:

Udi Manber, Introduction to Algorithms: A creative approach, Addison-Wesley.

## Course objectives:

Teach the basics of algorithms, computability and complexity to students who are not CSE majors

## Topics covered:

Review of Induction  
Basic Algorithm Design Techniques  
Graph Algorithms  
Fast Fourier Transform  
Turing machines and Computability  
Basics of Complexity  
NP-completeness & Reductions

## Class/laboratory schedule:

3 lectures per week. Bi-weekly assignments. Tests.

---

## Appendix I

(continued)

### C. Faculty CVs

- Anderson, Richard
- Anderson, Thomas
- Arnstein, Lawrence
- Baer, Jean-Loup
- Beame, Paul
- Borning, Alan
- Borriello, Gaetano
- Chambers, Craig
- Curless, Brian
- Dickey, Martin
- Diorio, Chris
- Domingos, Pedro
- Dooley, Emer
- Dugan, Benedict
- Ebeling, Carl
- Eggers, Susan
- Etzioni, Oren
- Fox, Dieter
- Gribble, Steven
- Halevy, Alon
- Karlin, Anna
- Kautz, Henry
- Ladner, Richard
- Lazowska, Ed
- Levy, Henry
- Mones, Barbara
- Notkin, David
- Perkins, Hal
- Popovic, Zoran
- Rao, Rajesh
- Ruzzo, Walter L.
- Salesin, David
- Seitz, Steven
- Sengupta, Rimli
- Shapiro, Linda
- Snyder, Lawrence
- Suciu, Dan
- Tanimoto, Steven
- Tompa, Martin
- Weld, Daniel
- Wetherall, David
- Zahorjan, John

# 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:

- Visiting Professor, Indian Institute of Science, Bangalore, India, 1993-1994

## Consulting, patents, etc.:

- Design Intelligence, 1997-1999

## Principal publications of last five years:

- Anderson, R.J., Tree data structures for N-body simulation, SIAM Journal on Computing, 28(6):1923--1940, 1999.
- Chan, W., Anderson, R.J., Beame, P., Burns, S., Modugno, F., Notkin, D., Reese, J., Model checking large software specifications, IEEE Transactions on Software Engineering, 24(7):498-520, July 1998.
- Anderson, R.J., Woll, H., Algorithms for the certified write all problem, SIAM Journal on Computing, 26, 5, October 1997, pp.1277--1283.
- Anderson, R. J., Chan, W., Beame, P., Notkin, D. Experiences with the Application of Symbolic Model Checking to the Analysis of Software Specifications," Proceedings of the Andrei Ershov Third International Conference on Perspectives of System Informatics, pp. 355-361, July, 1999.
- Anderson, R. J., Sobti, S., The table layout problem, COMPGIOM '99. Proceedings of the 15th ACM Symposium on Computational Geometry, pp.115--123, 1999.
- Chan, W., Anderson, R. J., Beame, P., Jones, D. H., Notkin, D., and Warner, W. E., Decoupling synchronization from local control for efficient symbolic model checking of statecharts, Proceedings of the 1999 International Conference on Software Engineering, pp.142-151, May, 1999.
- Chan, W., Anderson, R.~J., Beame, P., Notkin, D., Improving efficiency of symbolic model checking for state-based system requirements, 1998 ACM SIGSOFT Symposium on Software Testing and Analysis, pp.~102--112, March, 1998.
- Chan, W., Anderson, R.~J., Beame, P., Notkin, D., Combining constraint solving and symbolic model checking for a class of systems with non-linear constraints, Computer Aided Verification, 9th International Conference, CAV'97, June, 1997, pp. ~316--327.
- Borning, A., Anderson, R.~J., and Freeman-Benson, B., Indigo: a local propagation algorithm for inequality constraints, Proceedings of the 1996 ACM Symposium on User Interface Software and Technology, pp 129--136.

## Scientific and professional societies of which a member:

- ACM

## Honors and awards:

- 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:**

- 1999-2000 Department Executive Committee, Professional Master's Program Coordinator, Masters Admissions Committee (Chair), Tutored Video Instruction Coordinator, Faculty Recruiting Committee.
- 1998-1999 Professional Master's Program Coordinator, Masters Admissions Committee (Chair), Educational Technology and Distance Learning
- 1997-1998 Professional Masters Program Coordinator, Masters Admissions (Chair), Extension Liaison
- 1996-1997 Graduate Admissions (Chair), Department Executive Committee, Professional Masters Program Coordinator.
- 1996-1997 Graduate Admissions (Chair), Department Executive Committee, Professional Masters Program Coordinator.
- 1995-1996 Graduate Admissions (Chair), Department Executive Committee, Professional Masters Program Coordinator, Departmental Retreat Organizer.

**States in which registered:** None

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

---

# Anderson, Thomas

Professor (1997 Assoc, 2001 Full)

---

## Education:

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

## Other Experience:

- Computer Science Division, University of California, Berkeley.  
Assistant Professor, 1991 – 1996, Associate Professor, 1996 -- 1997.
- GenRad Incorporated (manufacturer of automated test equipment)  
Development Engineer/Senior Development Engineer, 1983 -- 1987.

## Consulting, patents, etc.:

- Digital Equipment Corporation Systems Research Center and Western Research Lab,  
Palo Alto, CA
- Navisoft Corporation, Santa Barbara, CA
- Xerox Palo Alto Research Center, Palo Alto, CA

## Principal publications of last five years:

- T. Anderson, B. Bershad, E. Lazowska, and H. Levy.  
"Thread Management for Shared-Memory Multiprocessors."  
Handbook for Computer Science.
- T. Anderson. "The Nachos System."  
Appeared as an appendix in,  
Operating Systems Concepts, 4th Ed., A. Silberschatz  
and P. Galvin. Addison-Wesley (November 1993).
- Drew Roselli, Jay Lorch, and Tom Anderson.  
"A Comparison of File System Workloads."  
To appear in Proceedings of the 2000 USENIX Technical Conference,  
New Orleans, June 2000.
- Neal Cardwell, Stefan Savage, and Tom Anderson.  
"Modeling TCP Latency."  
To appear in Proceedings of the 2000 IEEE Infocom Conference,  
Tel-Aviv, Israel, March 2000.
- Amit Aggarwal, Stefan Savage, and Tom Anderson.  
"Understanding the Performance of TCP Pacing."  
To appear in Proceedings of the 2000 IEEE Infocom Conference,  
Tel-Aviv, Israel, March 2000.
- Neil Spring, Maureen Chesire, Mark Berryman, Vivek Sahasranaman,  
Thomas Anderson, and Brian Bershad.  
"Receiver Based Management of Low Bandwidth Access Links."  
To appear in Proceedings of the 2000 IEEE Infocom Conference,  
Tel-Aviv, Israel, March 2000.
- Stefan Savage, Neal Cardwell, David Wetherall and Tom Anderson.  
"TCP Congestion Control with a Misbehaving Receiver."  
ACM Computer Communications Review, v 29, no 5, October 1999.
- Vahdat, M. Dahlin, T. Anderson and A. Aggarwal.  
"Active Names: Flexible Location and Transport of Wide-Area Resources."  
Proc. 1999 USENIX Symposium on Internet Technologies and Systems (USITS),  
October 1999.



- S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson.  
"The End-to-End Effects of Internet Path Selection."  
Proc. of the ACM SIGCOMM '99 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication}, September 1999.
- Stefan Savage, Neal Cardwell and Tom Anderson.  
"The Case for Informed Transport Protocols."  
Proceedings of the Seventh Workshop on Hot Topics in Operating Systems, Rio Rico, AZ, March 1999.
- S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan.  
"Detour: A Case for Intelligent Internet Routing and Transport."  
IEEE Micro, February 1999. Selected as an Award Paper at Proc. 1998 Hot Interconnects VI (August 1998).
- R. Wang, T. Anderson, and D. Patterson.  
"Virtual Log-Based File Systems for a Programmable Disk."  
Proc. of the 1999 USENIX Symposium on Operating Systems Design and Implementation, February 1999.
- N. McKeown and T. Anderson.  
"A Quantitative Comparison of Iterative Scheduling Algorithms for Input-Queued Switches." Computer Networks and ISDN Systems, Elsevier Science Publishing, 1998.
- Dennis Lee, Patrick Crowley, Jean-Loup Baer, Tom Anderson, and Brian Bershad.  
"Execution Characteristics of Desktop Applications on Windows NT."  
Proc. of the 25th Annual International Symposium on Computer Architecture (June 1998).
- R. Wang, A. Krishnamurthy, R. Martin, T. Anderson, and D. Culler.  
"Modeling Communication Pipeline Latency."  
Proc. of the 1998 SIGMETRICS Conference (June 1998).
- Vahdat and T. Anderson.  
"Transparent Result Caching."  
Proc. of the 1998 USENIX Technical Conference (June 1998).
- Ghormley, D. Petrou, S. Rodrigues, and T. Anderson.  
"SLIC: An Extensibility System for Commodity Operating Systems."  
Proc. of the 1998 USENIX Technical Conference (June 1998).
- R. Wang, T. Anderson, and M. Dahlin.  
"Experience with a Distributed File System Implementation."  
Software Practice and Experience (1998).
- Ghormley, D. Petrou, S. Rodrigues, A. Vahdat, and T. Anderson.  
GLUnix: A Global Layer UNIX for a Network of Workstations.  
Software Practice and Experience, Special Issue on Experience with Distributed Systems (1998).
- Vahdat, T. Anderson, and M. Dahlin.  
WebOS: Operating System Services for Wide Area Applications.  
Proc. of High Performance Distributed Systems (1998).
- Belani, A. Vahdat, T. Anderson, and M. Dahlin.  
CRISIS: A Wide Area Security Architecture.  
Proc. of the Seventh USENIX Security Symposium (January 1998).
- S. Chandra, M. Dahlin, B. Richards, R. Wang, T. Anderson, and J. Larus.  
"Experience with a Language for Writing Coherence Protocols."  
Proc. of the USENIX Conference on Domain-Specific Languages (October 1997).
- S. Savage, M. Burrows, G. Nelson, P. Solbavarro, and T. Anderson.  
Eraser: A Dynamic Race Detector for Multi-Threaded Programs."

- ACM Transactions on Computer Systems 16, 1 (Feb. 1998).  
 Selected as an **Award Paper** in Proc. of the Sixteenth ACM Symposium on Operating System Principles (October 1997), pp. 27-37.
- J. Neefe, A. Costello, D. Roselli, R. Wang, and T. Anderson.  
 "Improving the Performance of Log Structured File Systems With Adaptive Methods."  
Proc. of the Sixteenth ACM Symposium on Operating System Principles (October 1997).
  - S. Perissakis, C. Kozyrakis, T. Anderson, K. Asanovic, N. Cardwell, R. Fromm, K. Keeton, D. Patterson, R. Thomas, and K. Yelick.  
 "Scalable Processors in the Billion-Transistor Era: IRAM."  
IEEE Computer (September 1997).
  - R. Fromm, S. Periakakis, N. Cardwell, D. Patterson, T. Anderson, and K. Yelick.  
 "The Energy Efficiency of IRAM Architectures."  
Proc. of the 24th Annual International Symposium on Computer Architecture (June 1997).
  - R. Martin, A. Vahdat, D. Culler and T. Anderson.  
 "Effects of Communication Latency, Overhead and Bandwidth in a Cluster Architecture."  
Proc. of the 24th Annual International Symposium on Computer Architecture (June 1997).
  - Patterson, T. Anderson, N. Cardwell, R. Fromm, K. Keeton, C. Kozyrakis, R. Thomas, and K. Yelick.  
 "Intelligent RAM (IRAM): Chips that remember and compute."  
Proc. of the 1997 IEEE International Solid-State Circuits Conference (February 1997), pp. 224-225.
  - Yoshikawa, B. Chun, P. Eastham, A. Vahdat, T. Anderson, and D. Culler.  
 "Using Smart Clients to Build Scalable Services."  
Proc. of the 1997 USENIX Conference (January 1997).
  - S. Rodrigues, T. Anderson, and D. Culler. "High-Performance Local Area Communication with Fast Sockets." Proc. of the 1997 USENIX Conference (January 1997).

**Scientific and professional societies of which a member:**

- Associate Editor, ACM Transactions on Computer Systems, 1998-pres.
- Program Committee Member, ACM SIGCOMM'00 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, 2000.
- Program Committee Member, Third Symposium on Operating System Design and Implementation, 1999.
- Member, NSF Panel, Research Directions in Computer Networking, November 1998.
- Member, NSF Panel, Software Barriers to High Performance Computing, July 1998.
- Program Committee Member, Fourth High Performance Computer Architecture Conference (HPCA-4), 1998.
- Program Committee Member, First USENIX NT Symposium, 1997.
- Program Committee Member, Seventh ACM Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-VII), October 1996.

**Honors and awards:**

- 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.
- IBM Graduate Ph.D. Fellowship, 1989.

**Institutional and professional service in last five years:**

- Nachos, a software project for teaching undergraduate operating systems.  
<http://www.cs.washington.edu/homes/tom/nachos>
- Salsa, a software project for teaching undergraduate networking.  
<http://www.cs.washington.edu/education/courses/461/98wi/project>
- Lecture notes and homework assignments for undergraduate operating systems, approx. 300 pages. Spanish translation available.  
<http://www.cs.berkeley.edu/homes/tom/cs162sp96>

**States in which registered:** none

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

---

# Arnstein, Lawrence

*Research assistant professor (1999 Ass't)*

---

## **Education:**

- B.S., Computer Engineering, Case Western Reserve University, 1985
- M.S., Computer Engineering, Carnegie Mellon University, 1989
- Ph.D., Computer Engineering, Carnegie Mellon University, 1993

## **Other Experience:**

- Stratos Product Development Group, Seattle WA, 7/98-present  
Business Development and Electrical Engineering
- Cadence Design Systems, Inc. Yokohama, Japan  
Director of Program Management Department 7/94-7/98  
Major role in establishing Cadence's consulting services business unit in Japan.  
Highlights include: Managing successful multi-million dollar consulting projects to improve design processes for designing large scale integrated circuits. Selling and managing product design services for Cadence clients in Japan. Leading the development and adoption of new division level business processes for the quality and financial control of our growing consulting services business. Guiding culturally diverse and geographically distributed organizations. Developing new design automation methods and technologies through client funded consulting projects. Some of these results have been published in commercial conferences, and can now be found in standard products from Cadence.

## **Consulting, patents, etc.:**

- Quicksilver Technology, Inc. January 2001 to Present,  
Consulting on the design of compiler and synthesis technology for highly parallel/configurable architectures.

## **Principal publications of last five years:**

- UbiComp 2000 Submission: Labscape, Experiment Capture in the Biology Laboratory, Submitted April 2001.
- Arnstein, L., Sigurdsson, S., Franza, R., "Ubiquitous Computing in the Biology Laboratory," Journal of Laboratory Automation (JALA), Vol 6, no 1, March 2001.
- SuperComputing 2000 Panel, Convergence of the Extremes, with David Culler
- The Sixth Annual Workshop on Frontiers in Distributed Information Systems June 2000 (FDIS 2000): Invisible Computing in the Biology Laboratory, June 2000.
- CHI 2001 Workshop "Building the Ubiquitous Computing User Experience". presented "Breaking Down Barriers in the Biology Lab," April, 2001.

## **Scientific and professional societies of which a member:**

- IEEE
- ACM

## **Honors and awards:**

- National Science Foundation Graduate Research Fellow
- Reinberger Memorial Case Alumni Association Merit Scholar

## **Institutional and professional service in last five years:**

- Article Reviewer for Transactions on VLSI
- NSF Panel for CADRE'00
- Reviewer for Special Issue of IEEE Computer Magazine

**States in which registered:** none

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

---

# 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

## Principal publications of last five years:

- X.Qin and J.-L.Baer "A comparative study of conservative and optimistic trace-driven simulations" in 28th Annual Simulation Symposium, April 1995 pp. 42-50 (Best Paper Award)
- T.-F.Chen and J.-L. Baer "Effective Hardware-Based Data Prefetching for High-Performance Processors" IEEE TC, 44, 11, May 1995 pp. 609-623
- D.Lee, J.-L.Baer, B. Calder and D.Grunwald "Instruction cache fetch policies for speculative execution" 22nd Symp. on Computer Architecture, June 1995 pp. 357-367
- T. Romer, D.Lee, G. Volker, A. Wolman, W. Wong, J.-L. Baer, B. Bershad, and H. Levy "The structure and performance of interpreters", Proc. of 7th Int. Conf. on Architectural Support for Programming Languages and Operating Systems, Oct 1996, pp. 150-159
- X.Qin and J.-L. Baer "On the Use and Performance of Explicit Communication primitives in Cache Coherent Multiprocessor Systems" 3rd Symposium on High-Performance Computer Architecture, 1997, pp.182-193
- X.Qin and J.-L. Baer "A Performance evaluation of cluster-based Architectures", Proceedings of Sigmetrics, pp 237-247, 1997.
- D.Lee, P.Crowley, J.-L. Baer, T.Anderson and B.Bershad "Execution Characteristics of Desktop Applications on Windows NT" 25th Symp. on Computer Architecture, June 1998 pp. 27-38
- X.Qin and J.-L. Baer "Optimizing Software Cache-Coherent Cluster Architectures" Proceedings of SC98 (CD-ROM), Nov 1998
- P.Crowley and J.-L.Baer "Trace sampling for desktop applications on Windows NT" in Workload Characterization: Methodology and Case Studies, IEEE CS Press 1999
- P.Crowley, M.Fiuczynski, J.-L.Baer, and B.Bershad "Workloads for Programmable Network Interfaces" in Second Workshop on Workload Characterization Oct 1999
- D. Lee, J.-L. Baer, B. Bershad and T. Anderson "Reducing Start-up Latency in Web and Desktop Applications", Proc. 3rd Usenix Windows NT Symposium, June 1999
- P. VanVleet, E.Anderson, L.Brown, J.-L. Baer, and A. Karlin "Pursuing the Performance Potential of Dynamic Cache Lines" Proc. ICCD, Oct 1999 pp 528-537
- W.Wong and J.-L.Baer "Modified LRU Policies for Improving Second-level cache Behavior" 6th Symposium on High-Performance Computer Architecture, Jan 2000 pp 49-60

- P.Crowley, M.Fiuczynski, J.-L.Baer, and B.Bershad ``Characterizing Processor Architectures for ProgrammableNetwork Interfaces" ICS 2000, May 2000

**Scientific and professional societies of which a member:**

- IEEE Computer Society
- ACM (SIGARCH, SIGOPS)

**Honors and awards:**

- Guggenheim Fellowship,1979-1980
- IEEE Fellow 1992
- ACM Fellow 1997

**Institutional and professional service in last five years:**

- ACM SIGARCH Chair 1995-99
- ACM SGB (Sec-Large SIG Advisor) 1998-2000
- Area Editor Journal of Parallel and Distributed Computing 1987-present
- Associate Editor Computer Languages 1975-1999
- Program Committee Chair HPCA-4, 1998 + numerous PC memberships
- External Review Committee, Dept of CS Univ. of Minnesota 1998
- Chair Search Committee, Dean of Engineering 1995-1996
- ABET Review Coordinator for CSE 1996
- CSE Executive Committee 1996-98
- College of Engineering Faculty Governance team (Chair) 1998-1999
- UIF2 Review Committee 1999
- Faculty Senate 1999-2000

**States in which registered:** none

**Level of activity:** prof societies - high; 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:

- 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.:

- IBM Almaden Research, Summer 1997

## Principal publications of last five years:

- Paul W. Beame and Samuel R. Buss, Proof Complexity and Feasible Arithmetics, volume 29 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1998.
- Paul W. Beame and Toniann Pitassi. Propositional Proof Complexity: Past, Present, and Future. In Current Trends in Theoretical Computer Science Entering the 21st Century, G. Paun, G. Rozenberg and A Salomaa eds., World Scientific, 2001, 42-70.
- Paul Beame, Russell Impagliazzo, and Toniann Pitassi. Improved depth bounds for small distance connectivity. Computational Complexity, 7(4):325–345, 1998.
- Paul W. Beame, Allan Borodin, Prabhakar Raghavan, Walter L. Ruzzo, and Martin Tompa. A time-space tradeoff for undirected graph traversal by walking automata. SIAM Journal on Computing, 28(3):1051–1072, 1999.
- Paul W. Beame, Stephen A. Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. Journal of Computer and System Sciences, 57:3–19, 1998. Special issue of invited papers from 1995 STOC conference.
- W. Chan, R. J. Anderson, P. Beame, S. Burns, F. Modugno, D. Notkin, and J. D. Reese. Model check-ing large software specifications. IEEE Transactions on Software Engineering, 24(7):498–520, July 1998. Special section of invited papers from 4th Foundations in Software Engineering Conference.
- Paul Beame and Søren Riis. More on the relative strength of counting principles. In Paul W. Beame and Samuel R. Buss, editors, Proof Complexity and Feasible Arithmetics, volume 39 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 13–35. American Mathe-matical Society, 1998.
- Paul W. Beame, Faith E. Fich, and Rakesh Sinha. Separating the power of EREW and CREWPRAMs with small communication width. Information and Computation, 138(1):89–99, October 1997.
- Paul W. Beame, Russell Impagliazzo, Jan Kraj ícek, Toniann Pitassi, and Pavel Pudl ák. Lower bounds on Hilbert’s Nullstellensatz and propositional proofs. Proc. London Math. Soc., 73(3):1–26, 1996.
- Paul W. Beame and Toniann Pitassi. An exponential separation between the parity principle and the pigeonhole principle. Annals of Pure and Applied Logic, 80:197–222, 1996.
- Paul W. Beame, Allan Borodin, Prabhakar Raghavan, Walter L. Ruzzo, and Martin Tompa. Time-space tradeoffs for undirected graph traversal by graph automata. Information and Computation, 130(2):101–129, November 1996.



- Richard J. Anderson, Paul W. Beame, and Eric Brisson. Parallel algorithms for arrangements. *Algo-rithmica*, 15(2):104–125, 1996. Special issue from 1990 SPAA conference.
- Paul W. Beame and Toniann Pitassi. Propositional Proof Complexity: Past, Present, and Future. *Bulletin of the European Association for Theoretical Computer Science*, 65:66–89, June 1998. The Computational Complexity Column (ed. E. Allender).
- Dimitris Achlioptas, Paul W. Beame, and Michael Molloy. A sharp threshold in proof complexity. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, Crete, July 2001.
- Paul W. Beame, Russell Impagliazzo, and Ashish Sabharwal. The resolution complexity of independent sets in random graphs. in *Proceedings of the 16th Annual IEEE Conference on Computational Complexity*, June 2001.
- Paul W. Beame, Michael Saks, Xiadong Sun, and Erik Vee. Superlinear time-space tradeoff lower bounds for randomized computation. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 169-179, Redondo Beach, CA, November 2000. IEEE.
- W. Chan, R. J. Anderson, P. Beame, D. J. Jones, D. Notkin, and W. E Warner. Decoupling synchronization from logic for efficient symbolic model checking of statecharts. In *Proceedings of the 21st International Conference on Software Engineering*, pages 142–151, April 1999.
- Richard J. Anderson, William Chan, Paul W. Beame, and David Notkin. Experiences with the application of symbolic model checking to the analysis of software specifications. In *Proceedings of the Andrei Ershov Third International Conference on Perspectives of System Informatics*, pages 355–361, Novosibirsk, Russia.
- Paul W. Beame and Faith Fich. Optimal bounds for the predecessor problem. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 295–304, Atlanta, GA, May 1999.
- Paul W. Beame, Michael Saks, and Jayram S. Thathachar. Time-space tradeoffs for branching programs. In *Proceedings 39th Annual Symposium on Foundations of Computer Science*, pages 254–263, Palo Alto, CA, November 1998. IEEE.
- P. Beame, R. Karp, T. Pitassi, and M. Saks. On the complexity of unsatisfiability of random  $k$ -CNF formulas. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 561–571, Dallas, TX, May 1998.
- W. Chan, R. J. Anderson, P. Beame, and D. Notkin. Improving efficiency of symbolic model checking for state-based system requirements. In M. Young, editor, *ISSTA 98: Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 102–112, Clearwater Beach, FL, March 1998. Published as *Software Engineering Notes*, 23(2), March 1998.
- W. Chan, R. J. Anderson, P. Beame, and D. Notkin. Combining constraint solving and symbolic model checking for a class of systems with non-linear constraints. In O. Grumberg, editor, *Computer Aided Verification, 9th International Conference, CAV'97 Proceedings*, volume 1254 of *Lecture Notes in Computer Science*, pages 316–327, Haifa, Israel, June 1997. Springer-Verlag.
- Paul W. Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Proceedings 37th Annual Symposium on Foundations of Computer Science*, pages 274–282, Burlington, VT, October 1996. IEEE.
- R. J. Anderson, P. Beame, S. Burns, W. Chan, F. Modugno, D. Notkin, and J. D. Reese. Model checking large software specifications. In D. Garlan, editor, *Proceedings of the 4th ACM SIGSOFT Symposium on the Foundations of Software Engineering*, pages 156–166, San Francisco, CA, October 1996. Also published as *Software Engineering Notes*, 21(6), November 1996.

- Paul W. Beame, Russell Impagliazzo, and Toniann Pitassi. Improved depth bounds for small distance connectivity. In Proceedings 36th Annual Symposium on Foundations of Computer Science, pages 692–701, Milwaukee, WI, October 1995. IEEE.
- Paul W. Beame, Stephen A. Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. In Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, pages 303–314, Las Vegas, NV, May 1995.

**Scientific and professional societies of which a member:**

- Association for Computing Machinery

**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:**

- Co-chair, DIMACS Workshop on Inherent Complexity of Problems, April 2000
- Associate Editor, Computational Complexity
- Program Committee Chair, IEEE Symposium on Foundations of Computer Science, 1999
- Program Committee Member, Computational Complexity Conference, 1999
- NSF CCR-TOC CAREER Awards Panel, 1998
- Program Advisory Committee, Fields Institute special half-year on computational complexity, 1998.
- Program Committee Member, ACM Symposium on Theory of Computing, 1997
- Program Committee Member, International Computing and Combinatorics Conference, 1997
- Co-chair, DIMACS Workshop on Feasible Arithmetic and Proof Complexity, April 1996
- Steering committee, DIMACS special year on Logic and Algorithms 1995-6.
- University of Washington Faculty Senate, 1995-8

**States in which registered:** none

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

# 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 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
- Research Intern, Learning Research Group, Xerox Palo Alto Research Center 1976 -- 1979 and summer 1980

## Consulting, patents, etc.:

- Aldus Corporation, Seattle, Washington
- American Bell, Indianapolis, Indiana
- Apple Computer, Cupertino, California
- Atari Sunnyvale Research Center, Sunnyvale, California
- Axon Corporation, Seattle, Washington
- Corgraphics, Inc., Seattle, Washington
- Data I/O, Redmond, Washington
- Intel Corporation, Aloha, Oregon
- Object Technology International, Ottawa, Canada
- Rank Xerox EuroPARC, Cambridge, England
- Teklicon, Mountain View, California
- Tektronix Computer Research Laboratory, Beaverton, Oregon
- Xerox Palo Alto Research Center, Palo Alto, California

## Principal publications of last five years:

- Alan Borning, Richard Anderson, and Bjorn Freeman-Benson, "Indigo: A Local Propagation Algorithm for Inequality Constraints," Proceedings of the 1996 ACM Symposium on User Interface Software and Technology, November 1996, pages 129--136.
- Alan Borning, Kim Marriott, Peter Stuckey, and Yi Xiao, "Solving Linear Arithmetic Constraints for User Interface Applications," Proceedings of the 1997 ACM Conference on User Interface Software and Technology, October 1997, pages 87-96.
- Warwick Harvey, Peter J. Stuckey, and Alan Borning, "Compiling Constraint Solving using Projection," Proceedings of the Third International Conference on the Principles and Practice of Constraint Programming, October 1997, pages 491-505.
- Alan Borning, Richard Lin, and Kim Marriott, "Constraints for the Web," Proceedings of the 1997 ACM Multimedia Conference, November 1997, pages 173-182.
- Alan Borning and Bjorn Freeman-Benson, "Ultraviolet: A Constraint Satisfaction Algorithm for Interactive Graphics," CONSTRAINTS: An International Journal, Special Issue on Constraints, Graphics, and Visualization, Vol. 3 No. 1, April 1998, pages 9--32.

- Greg Badros, Alan Borning, Kim Marriott, and Peter Stuckey, "Constraint Cascading Style Sheets for the Web," Proceedings of the 1999 ACM Conference on User Interface Software and Technology, November 1999, pages 73--82.
- T.R.G. Green, A. Borning, T. O'Shea, M. Minoughan, and R. Smith, "The Stripetalk Papers: Understandability as a Language Design Issue in Object-Oriented Programming Systems," in Prototype Based Programming, James Noble, Antero Taivalsaari, and Ivan Moore (eds.), Springer-Verlag, 1998.
- L. Denise Pinnel, Matthew Dockrey, A.J. Bernheim Brush, and Alan Borning, "Design of Visualizations for Urban Modelling," in VisSym '00, Joint Eurographics - IEEE TCVG Symposium on Visualization.
- Alan Borning, Richard Lin, and Kim Marriott, "Constraint-based Document Layout for the Web," Multimedia Systems Journal, Vol 8 No 3, 2000.
- Greg Badros, Jojada J. Tirtowidjojo, Kim Marriott, Bernd Meyer, Will Portnoy, and Alan Borning, "A Constraint Extension to Scalable Vector Graphics," Proceedings of WWW10, Hong Kong, May 2001.
- Warwick Harvey, Peter Stuckey, and Alan Borning, "Fourier Elimination for Compiling Constraint Hierarchies," to appear in Constraints: An International Journal.

**Scientific and professional societies of which a member:**

- Association for Computing Machinery
- Computer Professionals for Social Responsibility

**Honors and awards:**

- Forsythe Memorial Award, March 1976. (This is a Stanford Computer Science Department award for exceptional service in teaching by a graduate student.)
- 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:**

- Organizing Committee Member, International Conferences on Constraint Programming, 1993--present.
- Program Committee, Constraints Stream, First International Conference on Computational Logic (conference held in London, July 2000).
- Program Committee, ACM Symposium on User Interface Software Technology (symposium held in San Diego, November 2000).
- Program Committee, Smart Graphics Symposium 2001, Conference held March 2001, Hawthorne, New York.
- Doctoral Program Committee, Seventh International Conference on Principles and Practice of Constraint Programming (CP 2001). Conference to be held November 2001, Paphos, Cyprus.
- 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.
- College of Engineering Physics Course Review Committee, Chair, 2000.
- Reviewing for National Science Foundation, Australian Research Council, numerous journals and conferences

**States in which registered:** none

**Level of activity:** prof 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

## Consulting, patents, etc.:

- Cadence Design Systems, 1998
- Georgia Board of Regents, 1999-2000
- Consystant Design Technologies (Chair, Technical Advisory Board), 1999-
- Intel, 2000
- Microsoft Research, 2000
- A. Bell, R. Lyon, G. Borriello. Self-Calibrated Clock and Timing Signal Generator for MOS/VLSI Circuitry, U. S. Patent 4,494,021; January 15, 1985.
- G. Borriello, R. Lyon, A. Bell. Data and Clock Recovery System for Data Communication Controller, U.S. Patent 4,513,427; May 1, 1985.
- C. Ebeling, G. Borriello. Dynamically Reconfigurable Logic Array for Digital Logic Circuits, U.S. Patent 5,208,491; May 1993.
- S. Hauck, G. Borriello, S. Burns, C. Ebeling. Field-Programmable Gate Array for Synchronous and Asynchronous Operation, U.S. Patent 5,367,209; November 22, 1994.

## Principal publications of last five years:

- T. Amon, G. Borriello, T. Hu, J. Liu. Symbolic Timing Verification of Timing Diagrams using Presburger Formulas. 34th ACM/IEEE Design Automation Conference, Anaheim, CA, pp. 226-231, June 1997.
- K. Hines, G. Borriello. Dynamic Communication Models in Embedded System Co-Simulation. 34th ACM/IEEE Design Automation Conference, Anaheim, CA, pp. 395-400, June 1997.
- S. Hauck, G. Borriello. An Evaluation of Bipartitioning Techniques. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 16, No. 8, pp. 849-866, August 1997.
- S. Hauck, G. Borriello. Pin-Assignment in Multi-FPGA Systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 16, No. 9, pp. 956-964, September 1997.
- T. Amon, G. Borriello, T. Hu, J. Liu. Making Complex Timing Relationships Readable: Presburger Formula Simplification Using Don't Cares. 35th ACM/IEEE Design Automation Conference, San Francisco, CA, pp. 586-590, June 1998.
- P. Chou, G. Borriello. Modal Processes: Towards Enhanced Retargetability through Control Composition of Distributed Embedded Systems, 35th ACM/IEEE Design Automation Conference, San Francisco, CA, pp. 88-93, June 1998.
- K. Hines, G. Borriello. A Geographically Distributed Framework for Embedded System Design and Validation, 35th ACM/IEEE Design Automation Conference, San Francisco, CA, pp. 140-145, June 1998.
- S. Hauck, G. Borriello, C. Ebeling. Mesh Routing Topologies for Multi-FPGA Systems. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 6, No. 3, PP. 400-408, September 1998. Best paper award.

- R. Ortega, G. Borriello. Communication Synthesis for Distributed Embedded Systems. IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, pp. 437-444, November 1998.
- P. Chou, K. Hines, K. Partridge, G. Borriello. Control Generation for Embedded Systems Based on Composition of Modal Processes. IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, pp. 46-53, November 1998.
- G. Borriello, L. Lavagno, R. Ortega. Interface Synthesis: a vertical slice from digital logic to software components. IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, November 1998.
- P. Chou, K. Hines, R. Ortega, K. Partridge, G. Borriello. ipChinook: An Integrated IP-based Design Framework for Distributed Embedded Systems, 36th ACM/IEEE Design Automation Conference, New Orleans, LA, June 1999.
- M. Esler, J. Hightower, G. Borriello. Next Century Challenges: Data-Centric Networking for Invisible Computing, ACM SIGMOBILE 5th International Conference on Mobile Computing and Networking, Seattle, WA, pp. 256-262, August 1999.
- G. Borriello. Capstone Design at the University of Washington: User Interfaces for Portable Devices. ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI2000), The Hague, The Netherlands, April 2000.
- G. Borriello, R. Want. Embedded Computation Meets the World-Wide-Web. Communications of the ACM, Vol. 43, No. 5, pp. 59-66, May 2000.
- R. Want, G. Borriello. Tutorial on Information Appliances. IEEE Computer Graphics & Applications, Vol. 20, No. 3, pp. 24-31, May/June 2000.
- P. Chou, G. Borriello. Synthesis and Optimization of Coordination Controllers for Distributed Embedded Systems, 37th ACM/IEEE Design Automation Conference, June 2000.
- R. Grimm, J. Davis, B. Hendrickson, E. Lemar, T. Anderson, B. Bershad, G. Borriello, D. Wetherall. one.world - Towards a System Architecture for Pervasive Computing, 4th Symposium on Operating Systems Design and Implementation (OSDI 2000), San Diego, CA, October 2000.
- G. Borriello. The Challenges to Invisible Computing, IEEE Computer, Integrated Engineering column, Vol. 33, No. 11, pp. 123-125, November 2000.
- K. Partridge, L. Arnstein, G. Borriello, T. Whitted. Fast Intrabody Signaling, 3rd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2000), Monterey, CA, December 2000.
- G. Borriello, B. Friedman, P. Kahn. Ubiquitous Computing: Technical, Psychological, and Value-Sensitive Integrations, CHI 2001 Workshop on Distributed and Disappearing User Interfaces in Ubiquitous Computing, Seattle, WA, April 2001.
- R. Grimm, J. Davis, B. Hendrickson, E. Lemar, A. Macbeth, S. Swanson, T. Anderson, B. Bershad, G. Borriello, S. Gribble, D. Wetherall. Systems Directions for Pervasive Computing, 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), Elmau/Oberbayern, Germany, May 2001.

**Scientific and professional societies of which a member:**

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

**Honors and awards:**

- Fulbright Pisa Chair Award - Scuola Superiore Sant'Anna, Pisa, Italia, 1995-1996
- Distinguished Teaching Award - University of Washington, 1995

**Institutional and professional service in last five years:**

- Member, NSF/CISE Advisory Board, 2001-
- Program chair, Ubiquitous Computing Conference, 2002

- Program co-chair, Ubiquitous Computing Dagstuhl Workshop (with H. Gellersen, F. Mattern), 2001
- Associate editor, IEEE Transactions on CAD of Integrated Circuits and Systems, 1997-1999
- Program chair, 1st Workshop on Software Engineering for Wearable and Pervasive Computing, 2000
- Program co-chair, 4th University of Washington/Microsoft Research Summer Institute on the Technologies of Invisible Computing, 1999
- Program chair, 5th ACM/IEEE International Workshop on Hardware/Software Co-design, 1997
- General chair, 6th ACM/IEEE International Workshop on Hardware/Software Co-design, 1998
- Steering committee, Intel Computing Continuum Conference, 2000
- Program committee member, International Symposium on Wearable Computing (ISWC), 1999-2000
- Program committee member, International Symposium on System Synthesis (ISSS), 1997-1999
- Program committee member, Architectural Support for Programming Languages and Operating Systems, 2000
- Provost's ad Hoc Committee on Faculty Responsibilities & Rewards, 1998
- Associate Chair for Educational Programs, 1998-2000

**States in which registered:** none

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

---



# Chambers, Craig

*Associate professor (1991 Ass't, 1997 Assoc)*

---

## Education:

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

## Principal publications of last five years:

- Markus Mock, Manuvir Das, Craig Chambers, and Susan Eggers. Dynamic Points-To Sets: A Comparison with Static Analyses and Potential Applications in Program Understanding and Optimization. In Proceedings of the ACM SIGPLAN/SIGSOFT Workshop on Program Analysis for Software Tools and Engineering (PASTE '01), Snowbird, Utah, June 2001.
- David Grove and Craig Chambers. An Assessment of Call Graph Construction Algorithms. Accepted to ACM Transactions on Programming Languages and Systems (TOPLAS).
- Jonathan Aldrich, Craig Chambers, Emin Gün Sirer, and Susan Eggers. Comprehensive Synchronization Elimination for Java. Accepted to Science of Computer Programming.
- Markus Mock, Craig Chambers, and Susan J. Eggers. Calpa: A Tool for Automating Selective Dynamic Compilation. In Proceedings of the 33rd Annual International Symposium on Microarchitecture (MICRO '00), Monterey, CA, December 2000.
- Curtis Clifton, Gary Leavens, Craig Chambers, and Todd Millstein. MultiJava: Modular Symmetric Multiple Dispatch and Open Classes for Java. In Proceedings of the 2000 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '00), Minneapolis, MN, October 2000.
- Brian Grant, Markus Mock, Matthai Philipose, Craig Chambers, and Susan Eggers. DyC: An Expressive Annotation-Directed Dynamic Compiler for C. Journal of Theoretical Computer Science (TCS), Vol. 248, Nos. 1-2, October 2000.
- Brian Grant, Markus Mock, Matthai Philipose, Craig Chambers, and Susan J. Eggers. The Benefits and Costs of DyC's Run-Time Optimizations. ACM Transactions on Programming Languages and Systems (TOPLAS), Vol. 22, No. 5, September 2000.
- Craig Chambers, William Harrison, and John Vlissides. A Debate on Language and Tool Support for Design Patterns. In Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '00), Boston, MA, January 2000.
- Craig Chambers and Weimin Chen. Efficient Multiple and Predicate Dispatching. In Proceedings of the 1999 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '99), Denver, CO, November 1999.
- Jonathan Aldrich, Craig Chambers, Emin Gün Sirer, and Susan Eggers. Eliminating Unnecessary Synchronization from Java Programs. In Proceedings of the 1999 International Symposium on Static Analysis (SAS '99), Venice, Italy, September 1999.
- Craig Chambers, Igor Pechtchanski, Vivek Sarkar, Mauricio Serrano, and Harini Srinivasan. Dependence Analysis for Java. In Proceedings of the 12th International Workshop on Languages and Compilers for Parallel Computing (LCPC '99), San Diego, CA, August 1999.
- Todd Millstein and Craig Chambers. Modular Statically Typed Multimethods. Accepted to Journal of Information and Computation. Earlier versions appeared in Proceedings of the European Conference on Object-Oriented Programming (ECOOP '99), Lisbon, Portugal, June 1999, and in Proceedings of the Sixth International Workshop on Foundations of Object-Oriented Languages (FOOL '99), San Antonio, TX, January 1999.

- Brian Grant, Matthai Philipose, Markus Mock, Craig Chambers, and Susan Eggers. An Evaluation of Staged Run-Time Optimizations in DyC. In Proceedings of the 1999 SIGPLAN Conference on Programming Language Design and Implementation (PLDI '99), Atlanta, GA, May 1999.
- Michael Ernst, Craig Kaplan, and Craig Chambers. Predicate Dispatching: A Unified Theory of Dispatch. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP '98), Brussels, Belgium, July 1998.
- Greg DeFouw, David Grove, and Craig Chambers. Fast Interprocedural Class Analysis. In Proceedings of the 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '98), San Diego, CA, January 1998.
- David Grove, Greg DeFouw, Jeffrey Dean, and Craig Chambers. Call Graph Construction in Object-Oriented Languages. In Proceedings of the 1997 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '97), Atlanta, GA, October 1997.
- Brian Grant, Markus Mock, Matthai Philipose, Craig Chambers, and Susan Eggers. Annotation-Directed Run-Time Specialization in C. In Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation (PEPM'97), Amsterdam, the Netherlands, June 1997.
- Craig Chambers and Gary Leavens. Type Soundness for an Object-Oriented Language with Multi-Methods, Block Structure, and Modules. In Proceedings of the Fourth International Workshop on Foundations of Object-Oriented Languages (FOOL '97), Paris, France, January 1997.
- Jeffrey Dean, Greg DeFouw, David Grove, Vassily Litvinov, and Craig Chambers. Vortex: An Optimizing Compiler for Object-Oriented Languages. In Proceedings of the 1996 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '96), San Jose, CA, October 1996.
- Joel Auslander, Matthai Philipose, Craig Chambers, Susan J. Eggers, and Brian N. Bershad. Fast, Effective Dynamic Compilation. In Proceedings of the 1996 SIGPLAN Conference on Programming Language Design and Implementation (PLDI '96), Philadelphia, PA, May 1996.
- Craig Chambers. Synergies Between Object-Oriented Programming Language Design and Implementation Research. Invited paper, Second International Symposium on Object Technologies for Advanced Software (ISOTAS '96), Kanazawa, Japan, March 1996.
- Craig Chambers, Susan J. Eggers, Joel Auslander, Matthai Philipose, Markus Mock, and Przemyslaw Paradyak. Automatic Dynamic Compilation Support for Event Dispatching in Extensible Systems. In Proceedings of the First Workshop on Compiler Support for System Software (WCSSS '96), Tucson, AZ, February 1996.
- Brian N. Bershad, Stefan Savage, Przemyslaw Paradyak, Emin Gün Sirer, Marc E. Fiuczynski, David Becker, Craig Chambers, and Susan Eggers. Extensibility, Safety and Performance in the SPIN Operating System. In Proceedings of the Fifteenth Symposium on Operating Systems Principles (SOSP '95), December 1995.
- Craig Chambers and Gary T. Leavens. Typechecking and Modules for Multi-Methods. In ACM Transactions on Programming Languages and Systems (TOPLAS), Vol. 17, No. 6, November 1995. An earlier version appeared in Proceedings of the 1994 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '94), Portland, OR, October 1994.

**Scientific and professional societies of which a member:**

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

**Institutional and professional service in last five years:**

- Program committee, 2001 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '01)
- General chair, International Symposium on Memory Management (ISMM '00)
- Program committee, International Conference on Functional Programming (ICFP '00)
- Program committee, Seventh International Workshop on Foundations of Object-Oriented Languages (FOOL '00)
- Program committee, 1999 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '99)
- Program committee, International Symposium on Static Analysis (SAS '99)
- Chair, program committee, 1998 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '98)
- Program committee, Workshop on Profile and Feedback-Directed Compilation. At the 1998 International Conference on Parallel Architectures and Compilation Techniques (PACT '98)
- Program committee, 1998 European Conference on Object-Oriented Programming (ECOOP '98)
- Organizing and program committee, Second International Workshop on Types in Compilation (TiC '98)
- Program committee, 1997 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '97)
- Program committee, Seventh International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '96)
- Program committee, 2nd International Symposium on Object Technologies for Advanced Software (ISOTAS '96)

**States in which registered:** none

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

---

# Curless, Brian

*Assistant professor (1998 Ass't)*

---

## 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:

- Editorial Board Member, Computers and Graphics, (1998 - present).
- Visiting Scientist, Digital Michelangelo Project, Florence, Italy (Winter 1999).
- Scientific Advisory Board Member, Paraform, Inc., Mountain View, CA (1998 - present).
- Software Contractor, Silicon Graphics Inc., Mountain View, CA (1993).
- Research Engineer, Remote Measurements Laboratory at SRI International, Menlo Park, CA (1988 - 1989).

## Principal publications of last five years:

- "Image Analogies," Aaron Hertzmann, Charles Jacobs, Nuria Oliver, Brian Curless, and David Salesin. To appear in SIGGRAPH'01.
- "The Digital Michelangelo Project: 3D scanning of large statues," Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. SIGGRAPH 2000.
- "Surface light fields for 3D photography," Daniel Wood, Daniel Azuma, Wyvern Aldinger, Brian Curless, Tom Duchamp, David Salesin, and Werner Steutzle. SIGGRAPH 2000.
- "Extensions to environment matting: towards higher accuracy and real-time capture," Yung-Yu Chuang, Douglas Zongker, Joel Hindorff, Brian Curless, David Salesin, and Rick Szeliski. SIGGRAPH 2000.
- "From range scans to 3D models," Brian Curless. Computer Graphics, November, 1999.
- "Environment matting and compositing," Douglas Zongker, Dawn Werner, Brian Curless, and David Salesin. SIGGRAPH '99, Los Angeles, CA, 9-13 August, 1999.
- "New Methods for Surface Reconstruction from Range Images," Brian Curless, Ph.D. dissertation, Stanford University, 1997.
- "A volumetric method for building complex models from range images," Brian Curless and Marc Levoy, SIGGRAPH '96, New Orleans, LA, 4-9 August 1996.
- "VripPack: Volumetric Range Image Processing Package," Brian Curless, software system available on the web (<http://www-graphics.stanford.edu/software/vrip>), 1996.
- "Better optical triangulation through spacetime analysis," Brian Curless and Marc Levoy, 1995 5th International Conference on Computer Vision, Boston, MA, 20-23 June 1995.

## Scientific and professional societies of which a member:

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

**Honors and awards:**

- Nominee for UW Distinguished Teaching Award (2001)
- ACM student nomination for best instructor (2000)
- CSE Department Packard Fellowship nominee (2000)
- Sloan Fellowship for Computer Science, University of Washington (2000)
- NSF CAREER Award, University of Washington (1999)
- Stanford Nominee for ACM Thesis Award, Stanford (1997)
- Achievement Rewards for College Scientists (ARCS) fellowship (1993)
- Gores Award for Teaching Excellence, Stanford (1992)
- Solid State Industrial Affiliates Fellowship, Stanford (1990)
- Graduated Summa Cum Laude, University of Texas (1988)

**Institutional and professional service in last five years:**

- 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:** prof societies - low; research - high; consulting/summer work in industry - low.

---

# Martin, Dickey

*Senior lecturer (1996 Lecturer, 2000 Senior Lecturer)*

---

## **Education:**

- B.A., Mathematics, Kent State University, 1969
- M.S., Mathematics, University of Kentucky, 1971
- Ph.D., Computer Science, Arizona State University, 1992

## **Other Experience:**

- 1972-1975 U.S. Maritime Administration, Washington, D.C.
- 1976-1977 Honeywell Information Systems, Washington, D.C.
- 1978-1986 Honeywell Information Systems, Phoenix, AZ
- 1989 (summer) Siemens (Munich)
- 1990 (spring) Grand Canyon University, Phoenix, AZ
- 1991-1996 Illinois College, Jacksonville, IL

## **Scientific and professional societies of which a member:**

- ACM
- ACL
- MAA

## **Institutional and professional service in last five years:**

- SIGCSE (Special Interest Group in Computer Science Education, Association for Computing Machinery): annual conference paper reviewer, 1998-2000.

**States in which registered:** none

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

---

# Diorio, Chris

*Associate professor (1997 Ass't, 2001 Assoc)*

---

## 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 November, 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 November, 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.
- B. A. Minch, P. Hasler, C. Diorio, and C. Mead, "An autozeroing floating-gate amplifier," U.S. Patent No. 5,875,126, issued 23 February, 1999.
- C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "A three-terminal silicon synaptic device," U.S. Patent No. 5,825,063, issued 26 July, 1998.
- C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "A semiconductor structure for long-term learning," U.S. Patent No. 5,627,392, issued 6 May, 1997.

## Principal publications of last five years:

- 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 Devices*, 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 derived 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 Devices*, 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 transistors," *Analog Integrated Circuits and Signal Processing*, vol. 9, no. 2, pp. 167-179, 1996.
- D. Hsu, M. Figueroa, and C. Diorio, "A silicon primitive for competitive learning," *Advances in Neural Information Processing Systems 13*, MIT Press, Cambridge, MA, 2001 (NIPS conference held Nov. 27-30 2000; proceedings in press).
- M. Richardson, J. Bilmes, and C. Diorio, "Hidden-articulator Markov models: Performance improvements and robustness to noise," *Proc. 6th Intl. Conf. on Spoken Language Processing*, Beijing, China, vol. 3, pp. 131-134, 2000.
- M. Richardson, J. Bilmes, and C. Diorio, "Hidden-articulator Markov models for speech recognition," *Proc. Automatic Speech Recognition: Challenges for the New Millennium*, Paris, France, pp. 133-139, 2000.
- D. Hsu, M. Figueroa, and C. Diorio, "A silicon primitive for competitive learning," *Proc. 4th Intl. Conf. on Cognitive and Neural Systems*, Boston, MA, May 25-27, 2000.
- T. Tuan, M. Figueroa, F. Lind, C. Zhou, C. Diorio, and J. Sahr, "An FPGA-based array processor for an ionospheric-imaging radar," *Proc. 2000 IEEE Symp. on Field-Programmable Custom Computing Machines*, Napa, CA, pp. 313-314, April 17-19, 2000.
- M. Figueroa and C. Diorio, "A 200MHz, 3mW, 16-tap mixed-signal FIR filter," *Proc. 2000 Symp. on VLSI Circuits*, Honolulu, HI, pp. 214-215, 2000.
- B. A. Minch, P. Hasler, and C. Diorio, "Synthesis of multiple-input translinear element networks," *Proc. 1999 IEEE Intl. Symp. on Circuits and Systems*, Orlando, FL, vol. 2, pp. 236-239, 1999.
- A. Pesavento, T. Horiuchi, C. Diorio, and C. Koch, "Adaptation of current signals with floating-gate circuits," *Proc. 7th Intl. Conf. on Microelectronics for Neural, Fuzzy, and Bio-Inspired Systems*, Granada, Spain, pp. 128-134, 1999.
- P. Hasler, B. A. Minch, and C. Diorio, "Adaptive circuits using pFET floating-gate devices," *Proc. 20th Anniversary Conf. on Advanced Research in VLSI (ARVLSI)*, Atlanta, GA, pp. 215-229, 1999.
- C. Zhou, F. D. Lind, T. Tuan, C. Diorio, and J. D. Sahr, "Realtime processor for estimating range-Doppler profiles of E-region irregularities with passive bistatic radar," a poster presented at the 1999 American Geophysical Union, San Francisco, CA, 1999.
- P. Hasler, B. A. Minch, and C. Diorio, "Floating-gate devices: They are not just for digital memories any more," *Proc. 1999 IEEE Intl. Symp. on Circuits and Systems*, Orlando, FL, vol. 2, pp. 388-391, 1999.
- P. Hasler, C. Diorio, and B. A. Minch, "A four-quadrant floating-gate synapse," *Proc. 1998 IEEE Intl. Symp. on Circuits and Systems*, Monterey, CA, vol. 3, pp. 29-32, 1998.



- P. Hasler, C. Diorio, and B. A. Minch, "Continuous-time feedback in floating-gate MOS circuits," Proc. 1998 IEEE Intl. Symp. on Circuits and Systems, Monterey, CA, vol. 3, pp. 90-93, 1998.
- B. A. Minch, P. Hasler, and C. Diorio, "The multiple-input translinear element: A versatile circuit element," Proc. 1998 IEEE Intl. Symp. on Circuits and Systems, Monterey, CA, vol. 1, pp. 527-530, 1998.
- P. Hasler, Andreas G. Andreou, Chris Diorio, Bradley A. Minch, and Carver Mead, "Impact ionization and hot-electron injection derived consistently from Boltzmann transport," Proc. 5th Intl. Workshop on Computational Electronics, Notre Dame, IN, May 28-30, 1997.
- B. A. Minch, P. Hasler, and C. Diorio, "Multiple-input translinear element networks," Proc. 1998 IEEE Intl. Symp. on Circuits and Systems, Monterey, CA, vol. 1, pp. 88-91, 1998.
- P. Hasler, B. A. Minch, and C. Diorio, "An autozeroing floating-gate bandpass filter," Proc. 1998 IEEE Intl. Symp. on Circuits and Systems, Monterey, CA, vol. 1, pp. 131-134, 1998.
- P. Hasler, T. Stanford, B. A. Minch, and C. Diorio, "An autozeroing floating-gate second-order section," Proc. 1998 IEEE Intl. Symp. on Circuits and Systems, Monterey, CA, vol. 2, pp. 351-354, 1998.
- C. Diorio, T. Humes, H. Notthoff, G. Chao, A. Lai, J. Hyde, M. Kintis, and A. Oki, "A 5.5GHz fractional frequency-synthesizer IC," Proc. 1997 IEEE GaAs IC Symp., Anaheim, CA, pp. 248-251, 1997.
- P. Hasler, B. A. Minch, C. Diorio, and C. Mead, "An autozeroing amplifier using pFET hot-electron injection," Proc. 1996 IEEE Intl. Symp. on Circuits and Systems, vol. 3, pp. 325-328, 1996.
- B. A. Minch, C. Diorio, P. Hasler, and C. Mead, "The matching of small capacitors for analog VLSI," Proc. 1996 IEEE Intl. Symp. on Circuits and Systems, vol. 1, pp. 239-241, 1996.
- B. A. Minch, C. Diorio, P. Hasler, and C. Mead, "A nMOS soft-maximum current mirror," Proc. 1995 IEEE Intl. Symp. on Circuits and Systems, vol. 3, pp. 2249-2252, 1995.
- P. Hasler, C. Diorio, B. A. Minch, and C. Mead, "Single transistor learning synapses with long term storage," Proc. 1995 IEEE Intl. Symp. on Circuits and Systems, vol. 3, pp. 1660-1663, 1995.
- P. Hasler, C. Diorio, B. A. Minch, and C. Mead, "Single transistor learning synapses," in Gerald Tesauro, David S. Touretzky, and Todd K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 817-824, MIT Press, Cambridge, MA, 1995.
- C. Diorio and R. P. N. Rao, "Neural circuits in silicon", *Nature*, vol. 405, no. 6789, pp. 891-892, 2000.
- C. Diorio, B. A. Minch, and P. Hasler, "Floating-gate MOS learning systems," in Proc. Intl. Symp. on the Future of Intellectual Integrated Electronics (ISFIIIE), Sendai, Japan, pp. 515-524, 1999.
- P. Hasler, B. A. Minch, J. Dugger, and Chris Diorio, "Adaptive circuits and synapses using pFET floating-gate devices," in G. Cauwenbergs and M. Bayoumi (eds.), *Learning in Silicon*, Boston, MA: Kluwer Academic Publishers, pp. 33-65, 1999.
- C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "Floating-gate MOS synapse transistors," in T. S. Lande (ed.), *Neuromorphic Systems Engineering: Neural Networks in Silicon*, Boston, MA: Kluwer Academic Publishers, pp. 315-337, 1998.
- C. Diorio, "Introduction: From neurobiology to silicon," in T. S. Lande (ed.), *Neuromorphic Systems Engineering: Neural Networks in Silicon*, Boston, MA: Kluwer Academic Publishers, pp. 263-266, 1998.

**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:**

- Panel member, NRC Committee on Frontiers at the Interface Between Computing and Biology. Sponsors: (1) NAS Computer Science and Telecommunications Board, and (2) DARPA Information Technology Office (2000-2001)
- Floating-gate workshop organizer and instructor at the 1999 NSF Telluride Workshop on Neuromorphic Engineering, held in Telluride, CO, from June 27 to July 17, 1999.
- Co-organizer, 1998 UW/MSR Summer Workshop on Intelligent Systems, held at the University of Washington from August 18-23, 1998.
- Floating-gate workshop organizer and instructor at the 1998 NSF Telluride Workshop on Neuromorphic Engineering, held in Telluride, CO, from June 28 to July 13, 1998.
- Floating-gate workshop organizer and instructor at the 1998 NSF Telluride Workshop on Neuromorphic Engineering, held in Telluride, CO, from June 24 to July 14, 1996.
- Floating-gate workshop organizer and instructor at the 1998 NSF Telluride Workshop on Neuromorphic Engineering, held in Telluride, CO, from June 26 to July 9, 1995.
- Departmental committees: 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

**States in which registered:** none

**Level of activity:** prof soc - medium; research - high; consulting/summer work in industry - medium.

# **Domingos, Pedro**

*Assistant professor (1999 Ass't)*

---

## **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:**

- "The Role of Occam's Razor in Knowledge Discovery", *Data Mining and Knowledge Discovery*, vol. 3, 1999.
- "Knowledge Discovery Via Multiple Models", *Intelligent Data Analysis*, vol. 2, 1998.
- "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss", first author, with Michael Pazzani, *Machine Learning*, vol. 29, 1997.
- "Context-Sensitive Feature Selection for Lazy Learners", *Artificial Intelligence Review*, vol. 11, 1997.
- "Unifying Instance-Based and Rule-Based Induction", *Machine Learning*, vol. 24, 1996.
- "Two-Way Induction", *International Journal on Artificial Intelligence Tools*, vol. 5, 1996.
- "Mining Time-Changing Data Streams," with Geoff Hulten and Laurie Spencer, in *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, 2001.
- "Mining the Network Value of Customers," with Matt Richardson, in *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, 2001.
- "A General Method for Scaling Up Machine Learning Algorithms and its Application to Clustering," in *Proceedings of the Eighteenth International Conference on Machine Learning*, Williamstown, MA, 2001.
- "Adaptive Web Navigation for Wireless Devices," with Corin Anderson and Daniel Weld, in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, WA, 2001.
- "Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach," with AnHai Doan and Alon Halevy, in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, Santa Barbara, CA, 2001.
- "Personalizing Web Sites for Mobile Users," with Corin Anderson and Daniel Weld, in *Proceedings of the Tenth International World Wide Web Conference*, Hong Kong, 2001.

- "Mixed Initiative Interfaces for Learning Tasks: SMARTedit Talks Back," with Steven Wolfman, Tessa Lau and Daniel Weld, in Proceedings of the Fifth International Conference on Intelligent User Interfaces, Santa Fe, NM, 2001.
- "A Unified Bias-Variance Decomposition for Zero-One and Squared Loss", in Proceedings of the Seventeenth National Conference on Artificial Intelligence, Austin, TX, 2000.
- "A Unified Bias-Variance Decomposition and its Applications," in Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, 2000.
- "Bayesian Averaging of Classifiers and the Overfitting Problem", in Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, 2000.
- "Version Space Algebra and its Application to Programming by Demonstration", with Tessa Lau and Daniel Weld, in Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, 2000.
- "Mining High-Speed Data Streams", first author, with Geoff Hulten, in Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining, Boston, MA, 2000.
- "MetaCost: A General Method for Making Classifiers Cost-Sensitive", in Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, San Diego, CA, 1999.
- "Process-Oriented Estimation of Generalization Error", in Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 1999.
- "Occam's Two Razors: The Sharp and the Blunt", in Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, 1998.
- "A Process-Oriented Heuristic for Model Selection", in Proceedings of the Fifteenth International Conference on Machine Learning, Madison, WI, 1998.
- "Why Does Bagging Work? A Bayesian Account and its Implications", in Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, 1997.
- "Knowledge Acquisition from Examples Via Multiple Models", in Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, 1997.
- "Linear-Time Rule Induction", in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, 1996.
- "Efficient Specific-to-General Rule Induction", in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, 1996.
- "Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier", first author, with Michael Pazzani, in Proceedings of the Thirteenth International Conference on Machine Learning, Bari, Italy, 1996.
- "Rule Induction and Instance-Based Learning: A Unified Approach", in Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995.

**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
- Program committee member, IJCAI-99
- Program committee member, ECML-2000
- NSF panel member (2000)
- Faculty mentor of students from African countries (1998-99)

**States in which registered:** none

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

---

# **Dooley, Emer**

*Lecturer (Lecturer 2000)*

---

**Education:**

- Ph.D., Business, University of Washington, 2000
- MBA, Strategic Management, University of Washington, 1992
- M. Eng, Computer Engineering, University of Limerick, Ireland, 1986
- B.S., Electronic Engineering, University of Limerick, Ireland, 1982

**Other Experience:**

- 1993-1994 Mosaix Inc. Marketing Manager.
- 1982-1989 Digital Equipment Corporation. Design Engineer.

**Honors and awards:**

- 1997 University of Washington Excellence in Teaching Award

**States in which registered:** none

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

---

# Dugan, Benedict

*Lecturer (Lecturer 1995)*

---

**Education:**

- B.S., Science & Technology Studies, Stanford University, 1992
- M.S., Computer Science, University of Washington, 1995

**Other Experience:**

- 2 years as TA/RA at Stanford and/or UW
- 3 years as Lecturer at UW
- 2 years as Chief Architect at VocalPoint Technologies Inc, San Francisco, CA

**States in which registered:** none

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

---

# 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

## **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. "A Configurable Architecture for Pipelined Computation", U.S. Patent issued 1999.

## **Principal publications of last five years:**

- Scott Hauck, Gaetano Borriello, Steven Burns, Carl Ebeling. The Triptych FPGA Architecture. IEEE Transactions on VLSI Systems, Vol. 3, No. 4, pp. 491-501, December 1995.
- Carl Ebeling, Larry McMurchie, Scott Hauck, Steven Burns. Placement and Routing Tools for the Triptych FPGA. IEEE Transactions on VLSI Systems, Vol. 3, No. 4, pp. 473-482, December 1995.
- Neil R. McKenzie, Carl Ebeling, Larry McMurchie and Gaetano Borriello. Experiences with the MacTester in Computer Science and Engineering Education. IEEE Transactions on Education, pp. 12-21, February 1997.
- Pak Chan, Martine Schlag, Carl Ebeling and Larry McMurchie. A Parallel Pathfinder Algorithm for Routing FPGAs. Accepted for publication, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.
- Carl Ebeling and Larry McMurchie. Pathfinder: A Negotiation-Based Router for Routing-Constrained FPGAs. In Proceedings of the Third ACM Symposium on Field-Programmable Gate Arrays, Monterey, pp. 111-117, February 1995.
- Brian Lockyear and Carl Ebeling. On the Performance of Level-Clocked Circuits. In Proceedings of the North Carolina Conference on Advanced Research in VLSI, Chapel Hill, pp. 342-356, March 1995.
- Soha Hassoun and Carl Ebeling. Architectural Retiming: Pipelining Latency-Constrained Circuits. In Proceedings of the Design Automation Conference, Las Vegas, pp. 708-713, June 1996.
- Soha Hassoun and Carl Ebeling. Using Precomputation in Architecture and Logic Resynthesis, In Proceedings of the International Conference on Computer-Aided Design, Santa Clara, November, 1998.
- Darren Cronquist, Paul Franklin, Miguel Figueroa, and Carl Ebeling. Architecture Design of Reconfigurable Pipelined Datapaths. In Proceedings of the Conference on Advanced Research in VLSI, Atlanta, April 1999.

## **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:**

- Chair, Program Committee, Fourth ACM Symposium on Field-Programmable Gate Arrays, February 1996.
- Member, Program Committee, International Conference on Computer Design, November, 1996.
- General Chair and Member of Program Committee, Fifth ACM Symposium on Field-Programmable Gate Arrays, February 1997.
- Member, Program Committee, 7th International Workshop on Field Programmable Logic and Applications, September 1997
- Member, Program Committee, 17th Conference on Advanced Research in VLSI, September 1997.
- Member, Program Committee, Sixth ACM Symposium on Field-Programmable Gate Arrays, February 1998.
- Guest editor, special issue on FPGAs, IEEE Transactions on VLSI Systems, April 1998
- Member, Program Committee, 8th International Workshop on Field Programmable Logic and Applications, September 1998
- Member, Program Committee, Seventh ACM Symposium on Field-Programmable Gate Arrays, February 1999
- Member, Program Committee, Reconfigurable Architectures Workshop, March 1999.
- Member, Program Committee, 9th International Workshop on Field Programmable Logic and Applications, September 1999.
- Member, Program Committee, Eighth ACM Symposium on Field-Programmable Gate Arrays, February 2000
- Associate Editor, IEEE Transactions on VLSI Systems, 1999-
- Graduate Program Coordinator, 1994-1998
- Faculty Senate, 1998-2000

**States in which registered:** none

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

---

# 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
- Thread Properties Attribute Vector Based Thread Selection in Multithreading Processor, CPQ Reference No.: PD96-0116

## Principal publications of last five years:

- J.L. Lo, S.S. Parekh, S.J. Eggers, H.M. Levy and D.M. Tullsen, Software-Directed Register Deallocation for SMT Processors, *IEEE Transactions on Parallel and Distributed Systems* (September 1999), pp. 922-933.
- J. Aldrich, C. Chambers, E.G. Sizer and S. Eggers, Static Analyses for Eliminating Unnecessary Synchronization from Java Programs, *Sixth International Static Analysis Symposium* (September 1999), pp. 19-38.
- B. Grant, M. Philipose, M. Mock, C. Chambers and S.J. Eggers, An Evaluation of Staged Run-time Optimizations, *Conference on Programming Language Design and Implementation* (May 1999), pp. 293-304.
- D.M. Tullsen, J.L. Lo, S.J. Eggers and H.M. Levy, Supporting Fine-Grain Synchronization on a Simultaneous Multithreaded Processor, *Fifth International Conference on High-Performance Computer Architecture* (January 1999), pp. 54-58.
- J.L. Lo, L. Barroso, S.J. Eggers, K. Gharachorloo, H.M. Levy and S.S. Parekh, An Analysis of Database Workload Performance on Simultaneous Multithreaded Processors, *International Symposium on Computer Architecture* (June, 1998), pp 39-50.
- J.L. Lo, S.J. Eggers, H.M. Levy, S.S. Parekh and D.M. Tullsen, Tuning Compiler Optimizations for Simultaneous Multithreading, *International Symposium on Microarchitecture* (December, 1997), pp. 114-124. Chosen for a special edition of *International Journal of Parallel Processing* (to appear).
- S.J. Eggers, J.S. Emer, H.M. Levy, J.L. Lo, R.L. Stamm and D.M. Tullsen, Simultaneous Multithreading: A Platform for Next-generation Processors, *IEEE Micro* (September/October, 1997), pp. 12-19.
- J.L. Lo, S.J. Eggers, J.S. Emer, H.M. Levy, R.L. Stamm and D.M. Tullsen, Converting Thread-level Parallelism into Instruction-level Parallelism via Simultaneous Multithreading, *ACM Transactions on Computer Systems* (August, 1997), pp. 322-354.
- B. Grant, M. Mock, M. Philipose, C. Chambers and S.J. Eggers, Annotation-directed Run-time Specialization in C, *Symposium on Partial Evaluation and Semantics-based Program Manipulation* (June, 1997), pp. 163-178.
- D.M. Tullsen, S.J. Eggers, J.S. Emer, H.M. Levy, J.L. Lo, and R.L. Stamm, Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor, *International Symposium on Computer Architecture* (May, 1996), pp 191-202.

- J. Auslander, M. Philipose, C. Chambers, S.J. Eggers and B.N. Bershad, Fast, Effective Dynamic Compilation, Conference on Programming Language Design and Implementation (May, 1996), pp. 149-159.

**Scientific and professional societies of which a member:**

- ACM
- IEEE
- AAUW
- AWIS

**Honors and awards:**

- 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:**

- Member, College Associate Dean Search: 1999/00
- Chair, Faculty Recruiting Committee: 1999/00
- University Disciplinary Committee, 1998/99
- Computer Science & Engineering Executive Committee: 1993/94, 1998/00.
- Affiliates Program Chair: 1994/95, 1995/96, 1997/98, 1998/99.
- Awards Committee: 1992/93, 1993/94, 1997/98.
- Graduate Student Recruiting: 1994/95, 1995/96.
- Graduate Student Evaluation Committee: 1993/94, 1994/95.

**States in which registered:** none

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

---

# Etzioni, Oren

*Associate professor (1991 Ass't, 1996 Assoc)*

---

## 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:

- Chief Technology Officer, Go2Net, Inc., May 1999 - April 2000
- Founder and Chief Scientist, Netbot, Inc., May 1996 - Present

## Consulting, patents, etc.:

- Chief Technology Officer, Go2Net, Inc., May 1999 - April 2000
- Founder and Chief Scientist, Netbot, Inc., May 1996 – Present, Netbot acquired an exclusive license to Ahoy!, Shopbot, and MetaCrawler from the University of Washington.
- The MetaCrawler service at [www.metacrawler.com](http://www.metacrawler.com) was sublicensed by Netbot to Go2net, Inc.
- The BRUTE data mining software has been licensed to:  
Boeing Computer Services, December 1994.  
Los Alamos National Laboratories, May 1995.  
Fermilab, May 1995.  
Hewlett Packard, January 1996.  
Over 20 licenses have been granted to universities including Vanderbilt University, the California Institute of Technology, the University of California at Irvine.
- The LCW closed-world reasoning software package was licensed on a non-exclusive basis to Apple, Inc. in 1996.

## Principal publications of last five years:

- A. Sugiura and O. Etzioni. Query Routing for Web Search Engines: Architecture and Experiments. In Proc. 9th Int. World Wide Web Conf., Amsterdam, The Netherlands, To appear.
- M. Perkowitz and O. Etzioni. Towards Adaptive Web Sites: Conceptual Framework and Case Study. Artificial Intelligence Journal. To appear.
- M. Perkowitz and O. Etzioni. Adaptive Web Sites: Concept and Case Study. Communications of the ACM. To appear.
- E. Selberg and O. Etzioni. On the Instability of Web Search Engines. In Proc. RIAO 2000.
- M. Perkowitz and O. Etzioni. Towards Adaptive Web Sites: Conceptual Cluster Mining. In Proc. 17th Int. Joint Conf. on AI, 1999.
- M. Perkowitz and O. Etzioni. Towards Adaptive Web Sites: Conceptual Framework and Case Study. In Proc. 8th Int. World Wide Web Conf., Toronto, Canada, 1999.
- O. Zamir and O. Etzioni. Grouper: A Dynamic Clustering Interface to Web Search Results. In Proc. 8th Int. World Wide Web Conf., Toronto, Canada, 1999.
- O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In Proc. SIGIR 1998, Melbourne, Australia, 1998.
- M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically Synthesizing Web Pages. In Proc. AAAI 1998, Madison, WI, 1998.
- D. Hsu, S. Soderland, and O. Etzioni. A redundant covering algorithm applied to text categorization. In Proc. 1998 AAAI/ICML Workshop on Learning for Text Categorization, 1998.

- A. Etzioni and O. Etzioni. Communities: Virtual vs. Real. *Science*, 277:295, July 1997.
- O. Zamir, O. Etzioni, O. Madani, and R. Karp. Fast and Intuitive Clustering of Web Documents. In Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, 1997.
- M. Perkowitz, R. Doorenbos, O. Etzioni, and D. Weld. Learning to understand information on the Internet: An example-based approach. *J. Intelligent Information Systems*, Vol.8 No.2, April 1997.
- M. Perkowitz and O. Etzioni. Adaptive web sites: an AI challenge. In Proc. 15th Int. Joint Conf. on AI, 1997.
- M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically learning from user access patterns. In Proc. 6th Int. World Wide Web Conf., Santa Clara, CA, 1997.
- J. Shakes, M. Langheinrich, and O. Etzioni. Dynamic reference sifting: a case study in the homepage domain. In Proc. 6th Int. World Wide Web Conf., 1997. Reprinted in *Computer Networks and ISDN Systems* 29 (1997) 1193-1204.
- R. Doorenbos, O. Etzioni, and D. Weld. A scalable comparison-shopping agent for the World Wide Web. In Proc. 1st Int. Conf. Autonomous Agents, pages 39-48, 1997.
- O. Etzioni, K. Golden, and D. Weld. Sound and efficient closed-world reasoning for planning. *Artificial Intelligence*, 89(1-2):113-148, January 1997.
- E. Selberg and O. Etzioni. The metacrawler architecture for resource aggregation on the web. *IEEE Expert*, 12(1):8-14, January 1997.
- O. Etzioni. Moving up the information food chain: softbots as information carnivores. In Proc. 13th National Conference Artificial Intelligence, 1996. Revised version reprinted in *AI Magazine special issue*, Summer 1997.
- O. Etzioni, S. Hanks, T. Jiang, and O. Madani. Optimal information gathering on the internet with time and cost constraints. Submitted to *SIAM Journal of Computing*., 1996.
- O. Etzioni, S. Hanks, T. Jiang, R. Karp, O Madani, and O. Waarts. Efficient information gathering on the internet. In *Foundations of Computer Science (FOCS)*, pages 234-243, 1996.
- N. Lesh and O. Etzioni. Scaling up goal recognition. In Proc. 5th Int. Conf. on Principles of Knowledge Representation and Reasoning, pages 178-189, 1996.
- O. Etzioni. The World Wide Web: quagmire or gold mine? *Communications of the ACM*, 37(7):65-8, 1996.
- E. Selberg and O. Etzioni. Multi-service search and comparison using the metacrawler. In Proc. 4th Int. World Wide Web Conf., pages 195-208, Boston, MA, 1995.
- N. Lesh and O. Etzioni. A sound and fast goal recognizer. In Proc. 14th Int. Joint Conf. on Artificial Intelligence, pages 1704-1710, 1995.
- M. Perkowitz and O. Etzioni. Category translation: Learning to understand information on the Internet. In Proc. 14th Int. Joint Conf. on Artificial Intelligence, pages 930-6, 1995.
- O. Etzioni, H. Levy, R. Segal, and C. Thekkath. The softbot approach to OS interfaces. *IEEE Software*, 12(4):42-51, 1995.
- O. Etzioni and D. Weld. Intelligent agents on the Internet: Fact, fiction, and forecast. *IEEE Expert*, 10(4):44-49, 1995.

**Scientific and professional societies of which a member:**

- American Association for Artificial Intelligence
- Association for Computing Machinery

**Honors and awards:**

- Edge Award for the best use of intelligent technology (in Jango) at the 1997 WebINNOVATION show.

- 1997 CommerceNet's New Innovator's Award for Jango.  
Best Paper Runnerup in the Sixth International World Wide Web Conference for Dynamic Reference Sifting: a Case Study in the Homepage Domain.
- 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 1997).
- MetaCrawler chosen as one of 3 finalists in the 1996 C|NET Awards for Best Internet Search Engine.
- Internet Softbot chosen as one of 5 finalists in the 1995 National DISCOVER Awards for Technological Innovation in Computer Software.
- NSF Young Investigator Award 1993.
- NSF Research Initiation Award 1992.
- AT&T Bell Laboratories Fellow 1987-90.

**Institutional and professional service in last five years:**

- President of AI Access Foundation
- Member of Advisory Board for Journal of Artificial Intelligence Research
- Guest editor, AI Magazine special issue on Intelligence on the Web (June 1997)
- Editorial Board: Journal of Artificial Intelligence Research, 1994-1996
- Editorial Board: Journal of Data Mining and Knowledge Discovery
- Editorial Board: for Springer Series in Agent Technology 1997
- International Conference on Autonomous Agents, 1999 (Program Chair)
- WWW8, 1998 (vice-chair for the "Searching, querying and indexing" track)
- First International Conference on Autonomous Agents, 1997 (Area Chair for Software Agents)
- International Conference on User Modeling, 1997 (program committee)
- American Association for Artificial Intelligence, 1996 (Senior member)
- Machine Learning Conference, 1995 (program committee)
- Second International Workshop on Next Generation Information Technologies and Systems 1995 (program committee)
- Lifelike Computer Characters 1994 and 1995 (program committee)
- AAAI Spring Symposium on Software Agents, 1994 (Chair)
- American Association for Artificial Intelligence, 1994 (program committee)

**States in which registered:** none

**Level of activity:** prof soc - medium; research - medium; consulting/summer work in industry - medium.

---

# Fox, Dieter

*Assistant professor (2000 Ass't)*

---

## 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

## Principal publications of last five years:

- S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz.  
Probabilistic algorithms and the interactive museum tour-guide robot minerva.  
In International Journal of Robotics Research (IJRR), 19 (11), 2000.
- D. Fox, W. Burgard, H. Kruppa, and S. Thrun.  
A Probabilistic Approach to Collaborative Multi-Robot Localization.  
Autonomous Robots, 8 (3), 2000.
- D. Schulz, W. Burgard, D. Fox, S. Thrun, and A.B. Cremers.  
Web interfaces for mobile robots in public places.  
IEEE-Magazine on Robotics and Automation, 7 (1), 2000.
- D. Fox.  
Markov Lokalisierung: Ein probalistisches Verfahren zur Lokalisierung und Navigation mobiler Roboter.  
Künstliche Intelligenz, 2000 (1) (in German, unrefereed).
- W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun.  
Experiences with an interactive museum tour-guide robot.  
Artificial Intelligence (AI), 114 (1-2) 1999.
- D. Fox, W. Burgard, and S. Thrun.  
Markov Localization for Mobile Robots in Dynamic Environments.  
Journal of Artificial Intelligence Research (JAIR), 11, 1999.
- D. Fox, W. Burgard, and S. Thrun.  
Active markov localization for mobile robots.  
Robotics and Autonomous Systems (RAS), 25:195-207, 1998.
- M. Betz, W. Burgard, D. Fox, and A.B. Cremers.  
Integrating active localization into high-level robot control systems.  
Robotics and Autonomous Systems (RAS), 23:205-220, 1998.
- S. Thrun, D. Fox, and W. Burgard.  
A probabilistic approach to concurrent mapping and localization for mobile robots.  
Machine Learning and Autonomous Robots (joint issue) (ML), (31), 1998.
- W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, A.M. Kappel, and S. Lüttringhaus-Kappel.  
Verbesserte Brandfrüherkennung im Steinkohlenbergbau durch Vorhersage von CO-Konzentrationen.  
KI Themenheft Data Mining, volume 1. ScienTec Publishing GmbH, 1998 (in German).
- D. Fox, W. Burgard, and S. Thrun.  
The dynamic window approach to collision avoidance.  
IEEE Robotics & Automation Magazine, 4(1), March 1997.

- J. Buhmann, W. Burgard, A.B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun.  
The mobile robot Rhino.  
AI Magazine, 16(2), Summer 1995.
- D. Fox, S. Thrun, F. Dellaert, and W. Burgard.  
Particle filters for mobile robot localization.  
In A. Doucet, N. de Freitas, and N. Gordon, editors, Sequential Monte Carlo Methods in Practice. Springer Verlag, New York, 2000.
- D. Fox.  
Markov Lokalisierung für mobile Roboter.  
In Ausgezeichnete Informatikdissertationen 1998, Teubner, Stuttgart, 2000 (in German, unrefereed).
- D. Fox, W. Burgard, H. Kruppa, and S. Thrun.  
Efficient Multi-Robot Localization Based on Monte Carlo Approximation.  
In J. Hollerbach and D. Koditschek, editors, Robotics Research: the Ninth International Symposium, Springer Verlag, 2000.
- S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt.  
Map learning and high-speed navigation in RHINO.  
In D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors, Artificial Intelligence and Mobile Robots. MIT/AAAI Press, Cambridge, MA, 1998.
- D. Schulz, W. Burgard, D. Fox, and A. B. Cremers.  
Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association.  
In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2001.
- S. Enderle, M. Ritter, D. Fox, S. Sablatnög, G. Kraetzschmar and G. Palm.  
Soccer-Robot Localization Using Sporadic Visual Features.  
In Proc. of the International Conference on Intelligent Autonomous Systems (IAS), 2000.
- R. Simmons, D. Apfelbaum, D. Fox, R. P. Goldman, K. Zita Haigh, D. J. Musliner, M. Pelican, and S. Thrun.  
Coordinated Deployment of Multiple, Heterogeneous Robots.  
In Proc. of the Conference on Intelligent Robots and Systems (IROS), 2000.
- S. Thrun, D. Fox, and W. Burgard.  
Monte carlo localization with mixture proposal distribution.  
In Proc. of the National Conference on Artificial Intelligence (AAAI), 2000.
- R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes.  
Coordination for multi-robot exploration and mapping.  
In Proc. of the National Conference on Artificial Intelligence (AAAI), 2000.
- S. Thrun, W. Burgard, and D. Fox.  
A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping.  
In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2000. Best paper award!
- W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun.  
Collaborative Multi-Robot Exploration.  
In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2000.
- D. Fox, W. Burgard, F. Dellaert, and S. Thrun.  
Monte carlo localization: Efficient position estimation for mobile robots.  
In Proc. of the National Conference on Artificial Intelligence (AAAI), 1999.
- D. Fox, W. Burgard, H. Kruppa, and S. Thrun.  
Collaborative Multi-Robot Localization.



- In Proc. of the German Conference on Artificial Intelligence (KI) and the 21st Symposium on Pattern Recognition (DAGM), Germany, 1999. DAGM outstanding paper award.
- S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, D. Schulz, and W. Steiner.  
Experiences with two deployed interactive tour-guide robots.  
In Proc. of the International Conference on Field and Service Robotics (FSR), 1999.
  - F. Dellaert, W. Burgard, D. Fox, and S. Thrun.  
Using the condensation algorithm for robust, vision-based mobile robot localization.  
In Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1999.
  - S. Thrun, J. Langford, and D. Fox.  
Monte carlo hidden markov models: Learning non-parametric models of partially observable stochastic processes.  
In Proc. of the International Conference on Machine Learning (ICML), 1999.
  - W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun.  
Sonar-based mapping of large-scale mobile robot environments using EM.  
In Proc. of the International Conference on Machine Learning (ICML), 1999.
  - S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz.  
MINERVA: A second generation mobile tour-guide robot.  
In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1999.
  - F. Dellaert, D. Fox, W. Burgard, and S. Thrun.  
Monte carlo localization for mobile robots.  
In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1999.
  - N. Roy, W. Burgard, D. Fox, and S. Thrun.  
Coastal navigation: Mobile robot navigation with uncertainty in dynamic environments.  
In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1999.
  - D. Fox, W. Burgard, S. Thrun, and A.B. Cremers.  
Position estimation for mobile robots in dynamic environments.  
In Proc. of the National Conference on Artificial Intelligence (AAAI), 1998.
  - S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers.  
Integrating topological and metric maps for mobile robot navigation: A statistical approach [ps.gz] (307 kb)  
In Proc. of the National Conference on Artificial Intelligence (AAAI), 1998.
  - W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun.  
The interactive museum tour-guide robot.  
In Proc. of the National Conference on Artificial Intelligence (AAAI), 1998.  
Outstanding paper award.
  - J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige.  
An experimental comparison of localization methods.  
In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1998. Best paper award.
  - W. Burgard, A. Derr, D. Fox, and A.B. Cremers.  
Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach.  
In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1998.

- D. Fox, W. Burgard, S. Thrun, and A.B. Cremers.  
A hybrid collision avoidance method for mobile robots.  
In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1998.
- S. Thrun, D. Fox, and W. Burgard.  
Probabilistic mapping of an environment by a mobile robot.  
In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1998.

**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:**

- 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.
- Outstanding paper award: 21st Symposium on Pattern Recognition (DAGM), 1999.
- Outstanding paper award: National Conference on Artificial Intelligence (AAAI), 1998.
- Best paper award: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1998.
- First place award: Clean-up tennis court event at the AAAI autonomous mobile robot competition, 1996.
- Second place award: Clean-up an office event at the AAAI autonomous mobile robot competition, 1994.

**Institutional and professional service in last five years:**

- Workshop on Interactive Robotics and Entertainment (WIRE), 2000.
- National Conference on Artificial Intelligence (AAAI): 1998, 1999, 2000.
- IEEE International Symposium on Computational Intelligence in Robotics & Automation (CIRA), 1999.
- Third European Workshop on Advanced Mobile Robots (EUROBOT), 1999.
- Co-chair of IJCAI-01 Workshop on Reasoning with Uncertainty in Robotics, 2001.
- Tutorial on Probabilistic Techniques for Mobile Robots. ICRA, 2001.
- Organizing committee of IJCAI-99 Workshop on Reasoning with Uncertainty in Robot Navigation, 1999.
- Editorial Board of Autonomous Robots (Special Issue on Integrating Robotics Research).

**States in which registered:** none

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

# Gribble, Steven

*Assistant professor (2000 Ass't)*

---

## 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

## Principal publications of last five years:

- What Can Peer-to-Peer Do for Databases, and Vice Versa?,  
by Steven Gribble, Alon Halevy, Zachary Ives, Maya Rodrig, and Dan Suciu.  
The Fourth International Workshop on the Web and Databases (WebDB '2001).
- Systems Directions for Pervasive Computing,  
by Robert Grimm, Janet Davis, Ben Hendrickson, Eric Lemar, Adam MacBeth,  
Steven Swanson, Tom Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble,  
and David Wetherall.  
The 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII).
- Robustness in Complex Systems,  
by Steven D. Gribble.  
The 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII).
- Scalable, Distributed Data Structures for Internet Service Construction,  
by Steven D. Gribble, Eric A. Brewer, Joseph M. Hellerstein, and David Culler.  
Proceedings of the Fourth Symposium on Operating Systems Design and  
Implementation (OSDI 2000).
- The Ninja Architecture for Robust Internet-Scale Systems and Services,  
by Steven D. Gribble, Matt Welsh, Rob von Behren, Eric A. Brewer, David Culler,  
N. Borisov, S. Czerwinski, R. Gummadi, J. Hill, A. Joseph, R.H. Katz, Z.M. Mao, S.  
Ross, and B. Zhao.  
To appear in a Special Issue of Computer Networks on Pervasive Computing.
- A Design Framework for Highly Concurrent Systems,  
by Matt Welsh, Steven D. Gribble, Eric A. Brewer, and David Culler.  
UC Berkeley CS Technical Report No. UCB/CSD-00-1108.
- The Ninja Jukebox,  
by Ian Goldberg, Steven D. Gribble, David Wagner, and Eric A. Brewer  
Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems,  
Boulder, CO, October 1999.
- The MultiSpace: an Evolutionary Platform for Infrastructural Services,  
by Steven D. Gribble, Matt Welsh, Eric A. Brewer, and David Culler  
Proceedings of the 1999 Usenix Annual Technical Conference.
- Experience With Top Gun Wingman: A Proxy-Based Graphical Web Browser for the  
3Com PalmPilot,  
by Armando Fox, Ian Goldberg, Steven D. Gribble, David C. Lee, Anthony Polito,  
and Eric A. Brewer  
Proceedings of Middleware '98, Lake District, England, September 1998.
- Self-Similarity in File Systems,  
by Steven D. Gribble, Gurmeet Singh Manku, Drew Roselli, Eric A. Brewer,  
Timothy J. Gibson, and Ethan L. Miller  
Proceedings of ACM SIGMETRICS '98, Madison, Wisconsin, June 1998.
- A Network Architecture for Heterogeneous Mobile Computing,  
by Eric A. Brewer, Randy H. Katz, Elan Amir, Hari Balakrishnan, Yatin Chawathe,  
Armando Fox, Steven D. Gribble, Todd Hodes, Giao Nguyen, Venkata N.

Padmanabhan, Mark Stemm, Srinivasan Seshan, and Tom Henderson  
IEEE Personal Communications.

- Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives,  
by Armando Fox, Steven D. Gribble, Yatin Chawathe, and Eric A. Brewer  
Special issue of IEEE Personal Communications on Adaption, August 1998.
- System Design Issues for Internet Middleware Services: Deductions from a Large Client Trace,  
by Steven D. Gribble and Eric A. Brewer  
Proceedings of the 1997 Usenix Symposium on Internet Technologies and Systems, Monterey, California, December 1997.
- Scalable Network Services,  
by Armando Fox, Steven D. Gribble, Yatin Chawathe, and Eric A. Brewer  
Proceedings of the 16th ACM Symposium on Operating Systems Principles (SOSP-16), St. Malo, France, October 1997.
- Orthogonal Extensions to the WWW User Interface Using Client-Side Technologies,  
by Armando Fox, Steven D. Gribble, Yatin Chawathe, Anthony S. Polito, Andrew Huang, Benjamin Ling, and Eric A. Brewer  
Demo at the 10th Annual Symposium on User Interface Software and Technology (UIST '97), Banff, Canada, October 1997.
- Adapting to Client Variability via On-Demand Dynamic Distillation,  
by Armando Fox, Steven D. Gribble, Eric A. Brewer, and Elan Amir  
Proceedings of the ACM Seventh International Conference on Architectural support for Programming Languages and Operating Systems, Cambridge, Massachusetts, October 1-5, 1996.
- Security on the Move: Indirect Authentication using Kerberos,  
by Armando Fox and Steven D. Gribble.  
Proceedings of the 2nd ACM International Conference on Mobile Computing and Networking (MobiCom '96) , Rye, New York, November 10-12, 1996.
- The Bay Area Research Wireless Access Network (BARWAN),  
by Randy H. Katz, Eric A. Brewer, Elan Amir, Hari Balakrishnan, Armando Fox, Steve Gribble, Todd Hodes, Daniel Jiang, Giao Thanh Nguyen, Venkata Padmanabhan, and Mark Stemm.  
Proceedings of the Spring COMPCON Conference 1996, Santa Clara, CA, February 25-28, 1996.
- Give and Take: Children Collaborating on One Computer,  
by Kori Inkpen, Kellogg S. Booth, Steven D. Gribble and Maria Klawe.  
Chi '95 Proceedings, Short Papers.

**Scientific and professional societies of which a member:**

- ACM
- IEEE
- USENIX

**Honors and awards:**

- ACM Teaching Award (May 2001)
- Cal @ Silicon Valley Fellowship, UC Berkeley CS, 1999.
- NSERC PGS-B Fellowship, 1997-1999, Canadian Federal Government.  
Canadian Space Agency NSERC Supplement, 1997-1999.
- NSERC PGS-A Fellowship, 1995-1997, Canadian Federal Government.  
Canadian Space Agency NSERC supplement, 1995-1997.
- Regent's Fellowship, UC Berkeley, 1995 (graduate program entrance award).
- Computing Research Association Outstanding Undergraduate Researcher, 1995 (honorable mention).

- Banks Scholarship, UBC, 1995.
- Rick Semple Memorial Award, UBC CS, 1995.
- McDonald Memorial Scholarship, UCB Physics, 1995.
- NSERC Undergraduate Summer Research Grant, UBC, 1993 and 1994.
- Science Scholar, UBC, 1993-1995, awarded to the top 20 students in the faculty of science.
- Canadian Information Processing Society Award, 1993.
- National Entrance Scholarship, UBC, 1990-1995, the top UBC entrance scholarship.
- Canada Scholarship, Canadian Federal Government, 1990-1995.

**Institutional and professional service in last five years:**

- Program Committee Member and WIP Organizer, The Third USENIX Symposium on Internet Technologies and Systems (USITS '01), March 2001.
- Publications and Publicity Chair, The Second IEEE Workshop on Internet Applications (WIAPP '01), July 2001

**States in which registered:** none

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

---

# Halevy, Alon

*Associate professor (1998 Ass't, 2001 Assoc)*

---

## Education:

- B.S., Computer Science and Mathematics, Hebrew University, Jerusalem, 1988
- Ph.D., Computer Science, Stanford University, 1993

## Other Experience:

- 4 and a half years at AT&T (Bell) Laboratories as a Principal Member of Technical Staff

## Consulting, patents, etc.:

- Founder of Nimble Technologies, a Seattle based company for providing advanced XML query processing tools.
- Author of 10 patents.

## Principal publications of last five years:

- Oliver M. Duschka, Michael R. Genesereth, Alon Y. Levy: Recursive Query Plans for Data Integration. JLP 43(1): 49-73 (2000)
- Zachary G. Ives, Daniela Florescu, Marc Friedman, Alon Y. Levy, Daniel S. Weld: An Adaptive Query Execution System for Data Integration. SIGMOD Conference 1999: 299-310
- Alon Y. Levy, Anand Rajaraman, Jeffrey D. Ullman: Answering Queries Using Limited External Query Processors. JCSS 58(1): 69-82 (1999)
- Alin Deutsch, Mary F. Fernandez, Daniela Florescu, Alon Y. Levy, Dan Suciu: A Query Language for XML. WWW8 / Computer Networks 31(11-16): 1155-1169 (1999)
- Mary F. Fernandez, Daniela Florescu, Jaewoo Kang, Alon Y. Levy, Dan Suciu: Catching the Boat with Strudel: Experiences with a Web-Site Management System. SIGMOD Conference 1998: 414-425.
- Alon Y. Levy, Marie-Christine Rousset: Verification of Knowledge Bases Based on Containment Checking. Artificial Intelligence 101(1-2): 227-250 (1998)
- Alon Y. Levy, Marie-Christine Rousset: Combining Horn Rules and Description Logics in CARIN. Artificial Intelligence 104(1-2): 165-209 (1998)
- Alon Y. Levy, Yumi Iwasaki, Richard Fikes: Automated Model Selection for Simulation Based on Relevance Reasoning. Artificial Intelligence 96(2): 351-394 (1997)
- Alon Y. Levy, Richard Fikes, Yehoshua Sagiv: Speeding up Inferences Using Relevance Reasoning: A Formalism and Algorithms. Artificial Intelligence 97(1-2): 83-136 (1997)
- Alon Y. Levy, Anand Rajaraman, Joann J. Ordille: Querying Heterogeneous Information Sources Using Source Descriptions. VLDB 1996: 251-262

## Scientific and professional societies of which a member:

- ACM
- IEEE

## Honors and awards:

- NSF Career Award – 2000
- Sloan Fellowship – 1999
- AAAI Best Paper Award - 1996

**Institutional and professional service in last five years:**

- Program committees:  
SIGMOD, VLDB -- 2001  
AAAI (senior PC member) 2000  
SIGMOD Panel Chair -- 2000  
PODS -1999  
SIGMOD - 1998  
AAAI - 1994, 96, 97, 98, 99.  
IEEE Data Engineering - 97,98,99
- Phd Admissions committee – 2000
- Faculty recruiting committee - 1999

**States in which registered:** none

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

---

# 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:

- Balanced Allocations, (with Y. Azar, A. Broder and E. Upfal).  
To appear in SIAM Journal on Computing.
- Implementation and Performance of Integrated Application-Controlled Caching, Prefetching and Disk Scheduling, (with P. Cao, E. Felten and K. Li).  
ACM Transactions on Computer Systems, Vol. 14, No. 4, November 1996.
- Markov Paging, (with S. Phillips and P. Raghavan).  
To appear in SIAM Journal on Computing.
- Near-Optimal Parallel Prefetching and Caching, with T. Kimbrel.  
To appear in SIAM Journal on Computing.
- Reducing Network Latency Using Subpages in a Global Memory Environment, (with H. Jamrozik, M. Feeley, G. Voelker, J. Evans, H. Levy and M. Vernon). 1996 International Conference on Architectural Support for Programming Languages and Operating Systems.
- A Trace-Driven Comparison of Algorithms for Parallel Prefetching and Caching, (with T. Kimbrel, A. Tomkins, R.H. Patterson, B. Bershad, P. Cao, E. Felten, G. Gibson, K. Li)  
In Proceedings of Second Symposium on Operating Systems Design and Implementation, October 1996.
- Implementing Cooperative Prefetching and Caching in a Globally-Managed Memory System, with G. Voelker, T. Kimbrel, E. Anderson, M. Feeley, J. Chase, and H. Levy. Proc. of the ACM SIGMETRICS Conf. on Measurement and Modelling of Computer Systems, June 1998.
- Organization-Based Analysis of Web-Object Sharing and Caching, with A. Wolman, G. Voelker, N. Sharma, N. Cardwell, M. Brown, T. Landray, D. Pinnell, and A. Karlin. Proc. of the 2nd USENIX Symp. on Internet Technologies and Systems (USITS), October 1999.
- On the scalability and performance of cooperative Web proxy caching, with A. Wolman, G. Voelker, N. Sharma, N. Cardwell, and A. Karlin. Proc. of the 17th ACM Symp. on Operating Systems Principles, December 1999.



- On List Update and Work Function Algorithms, with E. Anderson, K. Hildrum, A. Rasala and M. Saks.  
1999 European Symposium on Algorithms.
- Pursuing the Performance Potential of Dynamic Cache Line Sizes, with P. Van Vleet, E. Anderson and J.L. Baer.  
Proceedings of the 1999 International Conference on Computer Design.
- Random Walks with "Back Buttons", with R. Fagin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan.  
To appear Symposium on Theoretical Computer Science, 2000.
- Online Computation, (with S. Irani). Chapter in book  
"Approximation Algorithms for NP-Hard Problems",  
edited by Dorit Hochbaum, PWS Publishing Company, 1997.

**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:  
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
- Program Chair, IEEE Symposium on Foundations of Computer Science, 1997.
- Program Committee member:  
Ninth ACM Annual Symposium on Principles of Distributed Computing (1990),  
Twenty-fifth Annual ACM Symposium on Theory of Computing (1993),  
Fourth Annual ACM Symposium on Discrete Algorithms (1994),  
Sixth Annual ACM Symposium on Discrete Algorithms (1996),  
ACM Sigmetrics Conf. on Measurement and Modeling of Computer Systems (1997),  
ACM Symposium on Parallel Algorithms and Architectures} (1997),  
RANDOM'98: International Workshop on Randomization and Approximation  
Techniques in Computer Science} (1998),  
10th ACM-SIAM Symposium on Discrete Algorithms} (1999).

**States in which registered:** none

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

---

# Kautz, Henry

*Associate professor (2000 Assoc)*

---

## Education:

- B.A., Mathematics, Cornell University, 1978
- M.A., Creating Writing, Johns Hopkins University, 1980
- M.S., Computer Science, University of Toronto, 1982
- Ph.D., Computer Science, University of Rochester, 1987

## Other Experience:

- Technology Leader, AT&T Laboratories, 1997-2000.
- Department Head, AT&T Laboratories, 1994-1997.
- Senior Scientist, Bell Laboratories, 1987-1994.
- Systems Analyst, Nanodata Corporation, 1978-1979.

## Consulting, patents, etc.:

- "Optimization of information bases", US Patent issued Nov. 1993.
- "Mechanism for constraint satisfaction", US Patent issued June 1997.

## Principal publications of last five years:

- Balance and Filtering in Structured Satisfiable Problems. Henry Kautz, Yongshao Ruan, Dimitiri Achlioptas, Carla Gomes, Bart Selman, and Mark Stickel. Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001).
- Generating Satisfiable Problem Instances. Dimitris Achlioptas, Carla Gomes, Henry Kautz, and Bart Selman. Proc. AAAI-2000, Austin, TX.
- Hardening soft information sources. David McAllester, William Cohen, and Henry Kautz. Proc. KDD-2000, Boston, MA.
- Learning Declarative Control Rules for Constraint-Based Planning. Yi-Cheng Huang, Bart Selman, and Henry Kautz. Proc. ICML-2000 (Seventeenth International Conference on Machine Learning), Stanford, CA.
- Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. Carla P. Gomes, Bart Selman, Nuno Crator, and Henry Kautz. J. of Automated Reasoning, Vol. 24(1/2), pages 67-100, 1999.
- Unifying SAT-based and Graph-based Planning. Henry Kautz and Bart Selman. Proc. IJCAI-99, Stockholm, 1999.
- Integer Optimization Models of AI Planning Problems. Henry Kautz and Joachim P. Walser. Knowledge Engineering Review, vol. 15, no. 1, 2000, pp. 101-117.
- State-space Planning by Integer Optimization. Henry Kautz and Joachim P. Walser. Proc. AAAI-99, Orlando, FL, 1999.
- Control Knowledge in Planning: Benefits and Tradeoffs. Yi-Chen Huang, Bart Selman, and Henry Kautz. Proc. AAAI-99, Orlando, FL, 1999.
- BLACKBOX: A New Approach to the Application of Theorem Proving to Problem Solving. Henry Kautz and Bart Selman. Working notes of the Workshop on Planning as Combinatorial Search, held in conjunction with AIPS-98, Pittsburgh, PA, 1998.
- Creating Models of Real-World Communities with ReferralWeb. Henry Kautz and Bart Selman. Working notes of the Workshop on Recommender Systems, held in conjunction with AAAI-98, Madison, WI, 1998.
- Boosting Combinatorial Search Through Randomization. C.P. Gomes, B. Selman, and H. Kautz. Proc. AAAI-98, Madison, WI, 1998.

- The Role of Domain-Specific Knowledge in the Planning as Satisfiability Framework. Henry Kautz and Bart Selman. Proc. AIPS-98, Pittsburgh, PA, 1998. Abstract.
- Temporal Reasoning. Henry Kautz. Appears in the MIT Encyclopedia of Cognitive Science, forthcoming.
- A General Stochastic Approach to Solving Problems with Hard and Soft Constraints. Henry Kautz, Bart Selman, and Yueyen Jiang. In The Satisfiability Problem: Theory and Applications, Dingzhu Gu, Jun Du, and Panos Pardalos (Eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 35, American Mathematical Society, 1997, pages 573-586.
- Exploiting Variable Dependency in Local Search. Henry Kautz, David McAllester, and Bart Selman. DRAFT. Abstract appears in Abstracts of the Poster Sessions of IJCAI-97, Nagoya, Japan, 1997. Abstract.
- The Hidden Web. Henry Kautz, Bart Selman, and Mehul Shah. The AI Magazine, vol. 18, no. 2, Summer 1997, pages 27 - 36. In Adobe Acrobat format. Abstract.
- Ten Challenges in Propositional Reasoning and Search. Bart Selman, Henry Kautz, and David McAllester. Proc. IJCAI-97, Nagoya, Aichi, Japan, 1997.
- Temporal Reasoning. In The MIT Encyclopedia of Cognitive Science, MIT Press, Cambridge, 1999.
- Evidence for Invariants in Local Search. David McAllester, Bart Selman, and Henry Kautz. Proc. AAAI-97, Providence, RI, 1997. Abstract.
- ReferralWeb: Combining Social Networks and Collaborative Filtering. Henry Kautz, Bart Selman, and Mehul Shah. Comm. ACM, vol. 30 no. 3, March 1997.
- Encoding Plans in Propositional Logic. Henry Kautz, David McAllester, and Bart Selman, Proc. KR-96.
- Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search. Henry Kautz and Bart Selman, Proc. AAAI-96.

**Scientific and professional societies of which a member:**

- American Association for Artificial Intelligence (AAAI).
- Association for Computing Machinery (ACM).

**Honors and awards:**

- 1998 Elected to the Executive Council of the American Association for Artificial Intelligence
- 1997 Elected Fellow of the National Association for Artificial Intelligence
- 1996 Best Paper, 13th National Conference on Artificial Intelligence
- 1993 Best Paper, 11th National Conference on Artificial Intelligence
- 1989 Computers and Thought Award, by the International Joint Conference on Artificial Intelligence
- 1989 Best Paper, 1st International Conference on Knowledge Representation and Reasoning
- 1988 Best Paper, 7th Biennial Conference of the Canadian Society for the Computational Studies of Intelligence

**Institutional and professional service in last five years:**

- 2001 Advisory Committee, International Joint Conference on Artificial Intelligence
- 2001 Chair, LICS Workshop on Theory and Applications of Satisfiability Testing
- 2000 Program Co-Chair of the National Conference on Artificial Intelligence
- 2000 Reviewer for NASA Advanced Cross-Enterprise Technology Development
- 2000 Program Committee International Conference on Constraint Programming (CP2000)
- 1999 Associate Program Chair for the National Conference on Artificial Intelligence
- 1998 Chair AIPS-2000 Working on Planning as Combinatorial Search

- 1996-1998 Associate Editor of the Journal of Artificial Intelligence Research
- 1996-2001 Reviewer for the National Conference on Artificial Intelligence (AAAI) and the International Conference on Artificial Intelligence (IJCAI)

**States in which registered:** none

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

---

# 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:

- K.B. Erickson, R.E. Ladner, and A. LaMarca. Optimizing Static Calendar Queues. ACM Transactions on Modeling and Simulation, Vol. 10, No. 3, July 2000, 179-214.
- M.H. Johnson, R.E. Ladner, and E.A. Riskin. Fast Nearest Neighbor Search for Entropy-Constrained Vector Quantization. IEEE Transactions on Image Processing, Vol. 9, No. 8, 2000, 1435-1437.
- A.E. Mohr, E.A. Riskin, and R.E. Ladner. Unequal Loss Protection: Graceful Degradation of Image Quality over Packet Erasure Channels Through Forward Error Correction. IEEE Journal on Selected Areas in Communications, Vol. 18, No. 6, June 2000, 819-828.
- J. D. Fix and R.E. Ladner. Multiresolution Banded Refinement to Accelerate Surface Reconstruction from Polygons. Computational Geometry, Vol. 13, 1999, 49-64
- A. LaMarca and R.E. Ladner. The Influence of Caches on the Performance of Sorting. Journal of Algorithms, Vol. 31, 1999, 66-104.
- B.S. Srinivas, R. Ladner, E.A. Riskin, and M. Azizoglu. Progressive Transmission of Images using MAP Detection over Channels with Memory. IEEE Transactions on Image Processing, Vol. 8, No. 4, 1999, 462-475.
- J.D. Fix and R.E. Ladner. Sorting by Parallel Insertion on a One-Dimensional Sub-Bus Array. IEEE Transactions on Computers, Vol. 47, No. 11, 1998, 1267-1281.
- R.E. Ladner, A. LaMarca, and E. Tempero. Counting Protocols for Reliable End-to-End Transmission. Journal of Computer and System Sciences, Vol. 56, No. 1, 1998, 96-111.
- A. LaMarca and R.E. Ladner. The Influence of Caches on the Performance of Heaps. Journal of Experimental Algorithmics, Vol. 1. 1996.

- R.-Y. Wang, E.A. Riskin, and R. Ladner.  
Codebook Organization to Enhance Maximum A Posteriori Detection of Progressive Transmission of Vector Quantized Images over Noisy Channels.  
IEEE Transactions on Image Processing, Vol. 5, No. 1, 1996, 37-48.
- A. Condon, R.E. Ladner, J. Lampe, and R. Sinha.  
Complexity of Sub-Bus Mesh Computations.  
SIAM Journal on Computing, Vol. 25, No. 3, 1996, 520-539.
- E.S. Hong, R.E. Ladner, and E.A Riskin.  
Group Testing for Wavelet Packet Image Compression.  
IEEE Data Compression Conference (DCC 2001), 73-82.
- A. Bar-Noy and R.E. Ladner.  
Competitive On-Line Stream Merging Algorithms for Media-on-Demand.  
Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2001), 364-373. Revised and submitted for publication.
- A.E. Mohr, E.A. Riskin, and R.E. Ladner, Approximately Optimal Assignment for Unequal Loss Protection. ICIP 2000.
- E.S. Hong and R.E. Ladner.  
Group Testing for Image Compression.  
Proceedings IEEE Data Compression Conference (DCC 2000), pp. 3-12, March 2000.  
Revised and submitted for publication.
- A. E. Mohr, E. A. Riskin, and R. Ladner,  
Generalized Multiple Description Coding through Unequal Error Correction.  
Proceedings of ICIP 1999, vol. 1, pp. 411-415.
- A.E. Mohr, E.A. Riskin, and R.E. Ladner.  
Graceful Degradation over Packet Erasure Channels through Forward Error Correction.  
IEEE Data Compression Conference (DCC'99), pp. 92-101, March 1999.  
Revised and published.
- R.E. Ladner, J.D. Fix, and A. LaMarca.  
Cache Performance Analysis of Traversals and Random Accesses.  
Proceeding of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms.  
pp. 613-622.  
January 1999.
- K. Zatloukal, R.E. Ladner and M.H. Johnson.  
Nearest Neighbor Search in Vector Quantization.  
Workshop on Algorithm Engineering and Experimentation, January 1999. Revised and submitted to for AMS Series publication.
- J.D. Fix and R.E. Ladner.  
Multiresolution Banded Refinement to Accelerate Surface Reconstruction from Polygons.  
Fourteenth Annual ACM Symposium on Computational Geometry. pp. 240-248.  
1998. Revised and published.
- A. E. Mohr, E. A. Riskin, and R. Ladner. Bit Allocation for Wavelet Image Compression and Uniform Bit Loss.  
Proceedings of the 32th Annual Conference on Information Sciences and Systems, pp. 981-982, March 1998.
- A.E. Mohr, E. A. Riskin, and R. Ladner.  
Recovering from Bit Errors in Scalar-Quantized Discrete Wavelet Transformed Images .  
Proceedings of ICIP 1998, Volume III, pp. 502-506, October 1998.

- A. LaMarca and R.E. Ladner.  
The Influence of Caches on the Performance of Sorting.  
Eighth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 370-379, 1997.  
Revised and published.
- M.H. Johnson, R. Ladner, and E.A. Riskin.  
Fast Nearest Neighbor Search for ECVQ and other Modified Distorsion  
Measures by Comparing Boundary Distance.  
IEEE International Conference on Image Processing. pp. 423-426.  
September, 1996. Revised and published.

**Scientific and professional societies of which a member:**

- ACM and SIGACT
- IEEE Computer Society
- European Association for Theoretical Computer Science
- American Mathematical Society
- Association for Symbolic Logic

**Honors and awards:**

- Fullbright, 1992-93
- Guggenheim, 1985-86
- Fellow of ACM, 1994-

**Institutional and professional service in last five years:**

- External:  
Editor, Journal of Computer and System Sciences  
NSF Panel for Program in Theoretical Computer Science  
External Review Committee for Computer Sciences at Purdue University, 1999  
External Review Committee for Computer Science at CUNY, 1998  
Referee for several journals
- University:  
Member of Senate Planning and Budget Committee, 1996-1998
- College:  
Member of College of Engineering Promotion and Tenure Committee, 1995-1998
- Department:  
Chair of Faculty Recruiting, 1999  
Program Coordinator for Professional Masters Program, 2000  
Chair of Affiliates Program, 2000

**States in which registered:** none

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

---

# Lazowska, Ed

*Professor (1977 Ass't, 1982 Assoc, 1986 Full)*

---

## **Education:**

- A.B., Computer Science, 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

## **Consulting, patents, etc.:**

- Corporate Technical Advisory Boards:  
Microsoft Research, 1991-;  
Voyager Capital, 1996-;  
E-Quill, 2000-;  
Ignition, 2000-;  
Frazier Technology Ventures, 2000-;  
Madrona Venture Group, 2000-
- Corporate Boards of Directors:  
Data I/O Corp., 1996-;  
Lguide.com, 1999-

## **Principal publications of last five years:**

- T. Alberg, W. Finkbeiner, E. Lazowska, and D. Rosen. Policy Initiatives to Increase the Availability of Advanced Telecommunications Services Throughout Washington State. Technology Alliance of Washington, 1998.
- G. Voelker, H. Jamrozik, M. Vernon, H. Levy, and E. Lazowska. Managing Server Load in Global Memory Systems. Proc. ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, 1997.
- J. Chase, H. Levy, M. Feeley, and E. Lazowska. Sharing and Protection in a Single Address Space Operating Systems. ACM Transactions on Computer Systems 12,4 (1994).
- C. Thekkath, H. Levy, and E. Lazowska. Separating Data and Control Transfer in Distributed Operating Systems. Proc. 6th International Conference on Architectural Support for Programming Languages and Operating Systems, 1994.
- M. Vernon, E. Lazowska, and S. Personick, eds. R&D for the NII: Technical Challenges. EDUCOM, May 1994.

## **Scientific and professional societies of which a member:**

- National Academy of Engineering
- Association for Computing Machinery (Fellow)
- Institute of Electrical and Electronics Engineers (Fellow)
- IFIP Working Group 7.3 on Computer System Modeling

## **Honors and awards:**

- National Academy of Engineering, 2001
- Bill & Melinda Gates Endowed Chair in Computer Science, 2000
- BAM 100 (one hundred Brown University alumni who had the greatest impact on the twentieth century), 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
- University of Washington Annual Faculty Lecturer, 1996
- Fellow of the Institute of Electrical and Electronics Engineers, 1996
- Fellow of the Association for Computing Machinery, 1995
- Award paper, ACM SIGCOMM '93 Symposium
- 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
- Award Paper, 1985 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems
- Award paper, 7th ACM Symposium on Operating Systems Principles (1977)

**Institutional and professional service in last five years:**

Professional:

Information Science and Technology (ISAT) Study Group of the Defense Advanced Research Projects Agency, 1998-

Computer Science and Telecommunications Board of the National Research Council, 1996-

NSF Advisory Committee for Computer and Information Science and Engineering, 1995-2000 (Chair, 1998-2000)

Chair, Board of Directors, Computing Research Association, 1997-2001

Chair, ACM A.M. Turing Award Committee, 1999; Member, 1996-2001

National Science Foundation 50th Anniversary Public Advisory Committee, 1998-99

Networking Research Liaison Council, University Corporation for Advanced Internet Development, 1998-

Washington Software Alliance Board of Directors, 1996-

State of Washington Information Services Board, 1995-

Technology Alliance of Washington Council, 1996-; Board of Directors, 2000-; Vice President, 2001-

Chair, University-level review committees for Princeton Univ. Department of Computer Science (1999), Univ. of Virginia Department of Computer Science (1999), Rice Univ. Department of Computer Science (1998)

Standing Advisory Committees for UC Berkeley Department of EECS, Georgia Institute of Technology College of Computing, Hong Kong Univ. of Science and Technology

Dep't of Computer Science, National College of Ireland Program in Informatics

Institutional:

Chair, Department of Computer Science & Engineering, 1993-

Chair, University Advisory Committee on Academic Technology, 1990-

Search Committee for the Dean of the College of Education, 1999-00

Search Committee for Educational Technology Faculty, Coll. of Education, 1998-99

Chair, Committee to Review the Department of Statistics, 1998-99

Search Committee for the Head of the Graduate School of Library and Information

Sciences, 1997-98

Committee on the Future of the Graduate School of Library and Information Science,

1996-97

Presidential Task Force on Enrollment Planning, 1996-97

**States in which registered:** none

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

# 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 design team for VAX/VMS operating system, system architect for VAXclusters distributed system.

## **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.
- U.S. Patent. Thread Properties Attribute Vector Based Thread Selection in Multithreading Processors (with D. Tullsen, S. Eggers, J. Emer, T. Fossum, R. Stamm, R. Halstead and G. Chrysos). Approved January 2000. Filed by Compaq.

## **Principal publications of last five years:**

- Simultaneous Multithreading: Maximizing On-Chip Parallelism, with D. Tullsen and S. Eggers. Proc. of the 22nd International Symposium on Computer Architecture, June 1995. (Reprinted in G. Sohi, 25 Years of the International Symposia on Computer Architecture -- Selected Papers, ACM Press, 1998.)
- Implementing Global Memory Management in a Workstation Cluster, with M. Feeley, W. Morgan, F. Pighin, A. Karlin, and C. Thekkath. Proc. of the 15th ACM Symp. on Operating Systems Principles, December 1995.
- Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor, with D. Tullsen, S. Eggers, J. Emer, J. Lo, and R. Stamm, Proc. of the 23rd International Symposium on Computer Architecture, May 1996.
- Reducing Network Latency Using Subpages in a Global Memory Environment, with H. Jamrozik, M. Feeley, G. Voelker, J. Evans, A. Karlin, and M. Vernon. Proc. of the 7th ACM Symp. on Arch. Support for Prog. Languages and Operating Systems, Oct. 1996.
- Converting Thread-Level Parallelism to Instruction-Level Parallelism via Simultaneous Multithreading, with J. Lo, D. Tullsen, S. Eggers, J. Emer, and R. Stamm. ACM Trans. on Computer Systems, August 1997.
- Implementing Cooperative Prefetching and Caching in a Globally-Manged Memory System, with G. Voelker, T. Kimbrel, E. Anderson, M. Feeley, J. Chase, and A. Karlin.. Proc. of the ACM SIGMETRICS Conf. on Measurement and Modelling of Computing Systems, June 1998.
- An Analysis of Database Workload Performance on Simultaneous Multithreaded Processors, with J. Lo, S. Eggers, L. Barroso, K. Gharachorloo, and S. Parekh. Proc. of the 25th International Symposium on Computer Architecture, June 1998.

- Software-Directed Register Deallocation for Simultaneous Multithreaded Processors, with J. Lo, S. Parekh, S. Eggers, and D. Tullsen. IEEE Trans. on Parallel and Distributed Systems, September 1999.
- Manageability, Availability, and Performance in Porcupine: A Highly-Scalable Internet Mail Service, with Y. Saito and B. Bershad. Proc. of the 17th ACM Symposium on Operating Systems Principles, December 1999. (Award Paper.) Also, to appear, ACM Trans. on Computer Systems.
- On the Scaleability and Performance of Cooperative Web Proxy Caching, with A. Wolman, G. Voelker, N. Sharma, N. Cardwell, and A. Karlin. Proc. of the 17th ACM Symposium on Operating Systems Principles, December 1999

**Scientific and professional societies of which a member:**

- Fellow, Association for Computing Machinery
- Senior Member, IEEE

**Honors and awards:**

- Microsoft Professor (endowed professorship), 2000
- Fellow of ACM, 1995
- Fulbright Research Scholar Award, 1992
- Outstanding Paper Awards:  
ACM Symp. on Operating Systems Principles (1985, 1987, 1989, 1991, 1999)  
Int. Conf. on Measurement and Modeling of Comp. Systems (1989)  
3rd European Conf. on Object-Oriented Programming (1989)

**Institutional and professional service in last five years:**

- Program Chair:  
16th ACM Symp. on Operating Systems Principles  
4th IEEE Int. Workshop on Hot Topics in Operating Systems
- Program Committee:  
6th Int. Symp. on High Performance Computer Architecture  
15th ACM Symp. on Operating Systems Principles
- Associate Editor, ACM Trans. on Computer Systems
- Chairman, ACM SIGOPS (Special Interest Group in Operating Systems)
- Chairman, research evaluation review committee, INRIA France

**States in which registered:** none

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

---

# **Mones, Barbara**

*Senior Lecturer (Senior Lecturer 1999)*

---

## **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
- 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.

## **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

## **Institutional and professional service in last five years:**

- 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.
- Created and edited animation production video for 1999 CSE Graduation
- Contributed as needed for several outreach activities related to Animation on campus  
--  
including this year's Open House, 10 Yr review and others.
- Recommended and helped implement the redesign of LA2 and equipment procurement for the animation area.

**States in which registered:** none

**Level of activity:** prof 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:

- 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.:

- 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:

- Michael Ernst, Jake Cockrell, William G. Griswold and David Notkin. Dynamically Discovering Likely Program Invariants to Support Program Evolution. *IEEE Transactions on Software Engineering* 27,2, pp. 99-123 (February 2001). An earlier version appeared in *Proceedings of the 21st International Conference on Software Engineering*, pp. 213-224 (May 1999).
- William Chan, Richard J. Anderson, Paul Beame, David H. Jones, David Notkin, and William E. Warner. Optimizing Symbolic Model Checking for Statecharts. *IEEE Transactions on Software Engineering* 27,2, pp. 170-190 (February 2001).
- Gail C. Murphy, David Notkin, and Kevin Sullivan. Software Reflexion Models: Bridging the Gap Between Source and High-Level Models. *IEEE Transactions on Software Engineering* 27,4, pp. 364-380 (April 2001). An earlier version appeared in *Proceedings of the ACM SIGSOFT 95 Symposium on the Foundations of Software Engineering*, pp. 18-28 (October 1995).
- Michael Ernst, Adam Czeisler, William G. Griswold, and David Notkin. Quickly Detecting Relevant Program Invariants. *22nd International Conference on Software Engineering* (June 2000).
- William Chan, Richard J. Anderson, Paul Beame, David H. Jones, David Notkin and William E. Warner. Decoupling Synchronization from Local Control for Efficient Symbolic Model Checking of Statecharts. In *Proceedings of the 21st International Conference on Software Engineering*, pp. 142-151 (May 1999). An extended version will appear in *IEEE Transactions on Software Engineering* (2000).
- Michael Ernst, Jake Cockrell, William G. Griswold and David Notkin. Dynamically Discovering Likely Program Invariants to Support Program Evolution. In *Proceedings of the 21st International Conference on Software Engineering*, pp. 213-224 (May 1999). An extended version will appear in *IEEE Transactions on Software Engineering* (2000).
- Jurgen Dingel, David Garlan, Somesh Jha and David Notkin. Towards a Formal Treatment of Implicit Invocation using Rely/Guarantee Reasoning. *Formal Aspects of Computing* 10, pp. 193-213 (1998).

- Jurgen Dingel, David Garlan, Somesh Jha and David Notkin. Reasoning about Implicit Invocation. In Proceedings of the ACM SIGSOFT Sixth International Symposium on the Foundations of Software Engineering, pp. 209-221 (November 1998).
- Amir Michail and David Notkin. Illustrating Object-Oriented Library Reuse by Example: A Tool-Based Approach. In Proceedings of the 13th IEEE International Conference on Automated Software Engineering '98 (October 1998).
- William Chan, Richard Anderson, Paul Beame, Steve Burns, Francesmary Modugno, David Notkin, Jon Reese. Model Checking of Large Software Specifications. IEEE Transactions on Software Engineering 24, 7, pp. 498-520 (July 1998). An earlier version appeared in Proceedings of the Fourth ACM SIGSOFT Symposium on the Foundations of Software Engineering (October 1996).
- William Chan, Richard J. Anderson, Paul Beame and David Notkin. Improving Efficiency of Symbolic Model Checking for State-Based System Requirements. In ISSTA 98: Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (March 1998)
- Gail Alverson, William G. Griswold, Calvin Lin, David Notkin and Lawrence Snyder. Abstractions for Portable, Scalable Parallel Programming. IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 1, January 1998.
- Gail C. Murphy, David Notkin, William G. Griswold and Erica S.-C. Lan. An Empirical Study of Static Call Graph Extractors. ACM Transactions on Software Engineering and Methodology 7,2 (April 1998), pp. 158--191. An earlier version (Murphy, Notkin, Lan) appeared in Proceedings of the 18th International Conference on Software Engineering (March 1996).
- Gail C. Murphy and David Notkin. Reengineering with Reflexion Models: A Case Study. IEEE Computer 30,8 pp. 29-36 (August 1997). Translated (Japanese) and reprinted in Nikkei Computer, 19 (January 1998), pp. 161-169.
- William Chan, Richard. Anderson, Paul Beame, David Notkin, Combining Constraint Solving and Symbolic Model Checking for a Class of Systems with Non-linear Constraints. Proceedings of the 9th International Conference on Computer Aided Verification 97 (June 1997). Springer-Verlag Lecture Notes in Computer Science #1254. Orna Grumberg (Editor).
- Kingsum Chow and David Notkin. Semi-Automatic Update of Applications in Response to Library Changes. Proceedings of the International Conference on Software Maintenance '96, pp. 359-368 (November 1996).
- Michael VanHilst and David Notkin. Using Role Components to Implement Collaboration-Based Designs. OOPSLA-96 (October 1996).
- Michael VanHilst and David Notkin. Decoupling Change from Design In Proceedings of the Fourth ACM SIGSOFT Symposium on the Foundations of Software Engineering (October 1996).
- Kevin Sullivan, Ira J. Kalet and David Notkin. Mediators in a Radiation Treatment Planning Environment. IEEE Transactions on Software Engineering 22,8, pp. 563--579 (August 1996).
- Gail C. Murphy and David Notkin. On the Use of Static Typing to Support Operations on Frameworks. Object-Oriented Systems 3, 4, pp. 197-213 (April 1996).
- Carl Gunter, John Mitchell, and David Notkin. Strategic Directions in Software Engineering and Programming Languages. In ACM Computing Surveys, pp. 727-737, Vol. 28, Issue 4, (1996).
- Michael VanHilst and David Notkin. Using C++ Templates to Implement Role-Based Designs. In Second JSSST International Symposium on Object Technologies for Advanced Software, K. Futatsugi and S. Matsuoka (editors), pp. 22--37 (March 1996). Springer-Verlag Lecture Notes in Computer Science #1049.
- Gail C. Murphy and David Notkin. Lightweight Lexical Source Model Extraction. ACM Transactions on Software Engineering and Methodology 5,3, pp. 262-292 (July

1996). An earlier version appears in Proceedings of the ACM SIGSOFT 95 Symposium on the Foundations of Software Engineering, pp. 116--127 (October 1995).

- William G. Griswold and David Notkin. Architectural Tradeoffs for a Meaning-Preserving Program Restructuring Tool. IEEE Transactions on Software Engineering 21,4, pp. 275-287 (April 1995).

**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:**

- University of Washington Distinguished Graduate Mentor Award (2000).
- IBM Faculty Development Award (1985).
- NSF Presidential Young Investigator Award (1988).

**Institutional and professional service in last five years:**

- Associate Chair (2000-2001).
- Executive, Space & Visitors Committee (1995-97,1998-2000).
- CSE142/143 Stewardship and Community College Liaison (1999-2000).
- UW Educational Outreach Liaison (1995-2000).
- UW Branch Campus Liaison (1999-2000).
- Professional Masters Degree czar (1995-96).
- Executive Committee (1991-92, 1993-94, 1998-2000).
- Curriculum Committee (1986-87,96-97, chair 1991-95).
- University of Washington Tacoma, Computing and Software Systems, Faculty Recruiting Committee.
- University of Washington Bothell, Computing and Software Systems, Faculty Recruiting Committee.
- UW Educational Outreach Review Committee (1999).
- College of Engineering Promotion and Tenure Council (1998-2001).
- President's Advisory Committee for the School of Computer Science, Carnegie Mellon University (2000-2004).
- Board of Advisors, i5 Digital LLC (2000-).
- ACM Fellow (1998).
- Chair, Organizing Committee for the Workshop on Software Engineering and Programming Languages, Cambridge MA (June 1996).
- Co-chair, Working Group on Software Engineering and Programming Languages,
- CRA/ACM Strategic Directions in Computing Research, Cambridge MA (June 1996).
- Associate editor, IEEE Transactions on Software Engineering (1999- ).
- Associate editor, ACM Transactions on Software Engineering and Methodology (1989-1997).
- Associate editor, Journal of Programming Languages (1992-1997).
- Program Committee Chair, Co-chair, ICS2000: Software - Theory and Practice (part of
- IFIP World Computer Conference 2000), Beijing, August 2000.



- Program Committee Chair, Co-chair, 17th International Conference on Software Engineering (April 1995). Proceedings of the 17th International Conference on Software Engineering. Ross Jeffery and David Notkin, editors.
- Program Committee Chair, ACM SIGSOFT '93 Symposium on Foundations of Software Engineering (December 1993).
- ACM Software Systems Award Committee (1993-98, chair 1996-97).
- International Conference on Software Engineering Steering Committee (1993-2006, chair 1994-96).
- ACM SIGSOFT Chair (1997-2001).
- ACM SIGSOFT Secretary/Treasurer (1992-1997).
- ACM SIG Governing Board Executive Committee (1998-2001): Vice-Chair of Operations (1998-1999). Small SIG adviser (1999-2000).

**States in which registered:** none

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

---

# Perkins, Hal

*Senior lecturer (1998 Lecturer, 2001 Senior Lecturer)*

---

**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:** prof societies - low; research - none; consulting/summer work in industry - none.

---

# Popovic, Zoran

*Assistant professor (1999 Ass't)*

---

## **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:**

- 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)

## **Principal publications of last five years:**

- Jovan Popovic, Steven Seitz, Michael Erdmann, Zoran Popovic and Andy Witkin, "Interactive Manipulation of Rigid Body Simulations." in Computer Graphics (SIGGRAPH) 2000.
- Zoran Popovic, "Changing Physics for Character Animation." to appear in Communications of ACM (CACM), July 2000.
- Zoran Popovic "Editing Dynamic Properties of Captured Human Motion." in IEEE International Conference on Robotics and Automation (ICRA) 2000.
- Zoran Popovic and Andy Witkin "Physically Based Motion Transformation." in Computer Graphics (SIGGRAPH) 1999.
- Andy Witkin and Zoran Popovic "Motion Warping." in Computer Graphics (SIGGRAPH) 1995.

## **Scientific and professional societies of which a member:**

- ACM
- IEEE

## **Honors and awards:**

- Schlumberger Research Fellowship, 1995-99
- Baxter-Travenol Scholarship 1990-91 (merit-based full-tuition scholarship)

## **Institutional and professional service in last five years:**

- 2000 Graduate Student Admissions Committee
- 2000 Washington State Advanced Technology Initiative Center for Animation organizing committee
- 2000 Computer Animation 2000. Program Committee.
- 2000 SIGGRAPH'00 Course Organizer, Animating Humans by Combining Simulation and Motion Capture Organized a full day advanced level course on combining dynamics and motion capture for character animation
- 2000 SIGGRAPH'00 Course Speaker, Motion Editing: Principles, Practice and Promise Full day advanced level course on the current state of motion editing algorithms and how they are used in production
- 1999 SIGGRAPH'99 Course Speaker, Motion Editing: Principles, Practice and Promise Full day advanced level course on the current state of motion editing algorithms and how they are used in production.

**States in which registered:** none

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

---

# Rao, Rajesh

*Assistant 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

## Other Experience:

- Prepared lesson plans and delivered two lectures for an undergraduate course on Computational Neurobiology (BIPN 146) at University of California, San Diego, 1999. Professor: T. Sejnowski.  
Textbook: Biophysics of Computation by Christof Koch.
- Teaching Assistant, Department of Computer Science, University of Rochester, Spring 1993 and 1994. Courses: 1. Theory of Computation  
2. Design and Analysis of Algorithms.
- Teaching Assistant, Mathematics Department, Angelo State University, 1989-1992. Undergraduate courses on calculus and analytical geometry.
- Teaching Assistant, Physics Department, Angelo State University, 1989-1990. Undergraduate courses on fundamentals of physics.

## Principal publications of last five years:

- Rajesh P. N. Rao, David Eagleman, and Terrence J. Sejnowski. "Optimal Smoothing in Visual Motion Perception" *Neural Computation*, in press, 2001.
- Rajesh P. N. Rao and Terrence J. Sejnowski. "Spike Timing Dependent Hebbian Plasticity as Temporal Difference Learning" *Neural Computation*, in press, 2001.
- Rajesh P. N. Rao and Terrence J. Sejnowski. "Predictive Learning of Temporal Sequences in Recurrent Neocortical Circuits" 2001 *Complexity in Biological Info. Processing*, in press, 2001.
- Chris Diorio and Rajesh P. N. Rao, "Neural Circuits in Silicon" *Nature*, Vol. 405, pp. 891-892, 2000.
- Rajesh P. N. Rao and Dana H. Ballard. "Predictive Coding in the Visual Cortex: A Functional Interpretation of Some Extra-Classical Receptive Field Effects" *Nature Neuroscience*, Vol. 2(1), pp. 79-87, 1999.
- Rajesh P. N. Rao. "An Optimal Estimation Approach to Visual Perception and Learning" *Vision Research*, Vol. 39(11), pp. 1963-1989, 1999.
- Rajesh P. N. Rao and Daniel L. Ruderman. "Learning Lie Groups for Invariant Visual Perception" M. S. Kearns, S. A. Solla and D. Cohn (Eds.), *Advances in Neural Information Processing Systems 11*, Cambridge, MA: MIT Press, pp. 810-816, 1999.
- Rajesh P. N. Rao and Dana H. Ballard. "Development of Localized Oriented Receptive Fields by Learning a Translation-Invariant Code for Natural Images" *Network: Computation in Neural Systems*, Vol. 9(2), pp. 219-234, 1998.
- Rajesh P. N. Rao. "Correlates of Attention in a Model of Dynamic Visual Recognition" M. I. Jordan, M. J. Kearns and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems 10*, Cambridge, MA: MIT Press, pp. 80-86, 1998.

- Rajesh P. N. Rao and Olac Fuentes. "Hierarchical Learning of Navigational Behaviors in an Autonomous Robot using a Predictive Sparse Distributed Memory" *Autonomous Robots*, Vol. 5, pp. 297-316, 1998 and *Machine Learning*, Vol. 31, pp. 87-113, 1998.
- Rajesh P. N. Rao and Dana H. Ballard. "Dynamic Model of Visual Recognition Predicts Neural Response Properties in the Visual Cortex" *Neural Computation*, Vol. 9, pp. 721-763, 1997.
- Rajesh P. N. Rao. "Dynamic Appearance-Based Recognition" *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 540-546, 1997.
- Dana H. Ballard, Mary M. Hayhoe, Polly K. Pook, and Rajesh P.N. Rao. "Deictic Codes for the Embodiment of Cognition" *Behavioral and Brain Sciences*, Vol. 20(4), pp. 723-767, 1997.
- Gregory J. Zelinsky, Rajesh P. N. Rao, Mary M. Hayhoe, and Dana H. Ballard. "Eye Movements Reveal the Spatiotemporal Dynamics of Visual Search" *Psychological Science*, Vol. 8(6), pp. 448-453, 1997.
- Rajesh P. N. Rao, Gregory J. Zelinsky, Mary M. Hayhoe, and Dana H. Ballard. "Modeling Saccadic Targeting in Visual Search" D. Touretzky, M. Mozer and M. Hasselmo (Eds.), *Advances in Neural Information Processing Systems 8*, Cambridge, MA: MIT Press, pp. 830-836, 1996.

**Scientific and professional societies of which a member:**

- Society for Neuroscience

**Honors and awards:**

- Alfred P. Sloan Research Fellow, 2001-2003.
- Alfred P. Sloan Postdoctoral Fellow, Salk Institute for Biological Studies, 1997-2000.
- Computer Science Research and Teaching Assistantship, University of Rochester, 1992-1997.
- Presidential Fellowship for Graduate Studies, State University of New York, Buffalo, 1992 (declined in favor of Rochester).
- Robert and Nona Carr Academic Scholarship for undergraduate study, Angelo State University, 1988-1992.
- 1991 Who's Who among students in American Colleges and Universities.
- Alpha Chi (National Honor Scholarship), Epsilon Delta Pi (Computer Science), and Pi Mu Epsilon (Mathematics) 1991-1992.

**States in which registered:** none

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

---

# 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:

- G. Barnes and W. L. Ruzzo, "Undirected s-t Connectivity In Polynomial Time and Sublinear Space," *Computational Complexity*, 6(1):1-28, 1996-1997. Also appeared in *Proc. of the 23rd ACM Symp. on Theory of Computing*, New Orleans, LA, May 1991, pages 43-53.
- K. Chandra, P. Raghavan, W. L. Ruzzo, R. Smolensky and P. Tiwari, "The Electrical Resistance of a Graph Captures its Commute and Cover Times," *Computational Complexity*, 6(4):312-340, 1997. Also appeared in *Proc. of the 21st ACM Symp. on Theory of Computing*, Seattle, WA, May 1989, 574-586.
- G. Barnes, J. F. Buss, W. L. Ruzzo, B. Schieber, "A Sublinear Space, Polynomial Time Algorithm for Directed s-t Connectivity," *SIAM Journal on Computing*, 27(5):1273-1282, Oct. 1998.
- P. Beame, A. Borodin, P. Raghavan, W. L. Ruzzo, and M. Tompa, "A Time-Space Tradeoff for Undirected Graph Traversal by Walking Automata," *SIAM Journal on Computing* 28(3):1051-1072, 1999.
- W. L. Ruzzo and M. Tompa. "A linear time algorithm for finding all maximal scoring subsequences," *Proc. 7th Internatl Conf on Intelligent Systems for Molecular Biology*, 234-241, Heidelberg, Germany, Aug. 1999. AAAI Press.
- P. W. Dymond and W. L. Ruzzo, "Parallel Random Access Machines With Owned Global Memory and Deterministic Context-Free Language Recognition," *Journal of the ACM*, 47(1):16-45, Jan. 2000.
- J. Redstone and W. L. Ruzzo, "Algorithms for a Simple Point Placement Problem." Invited to Fourth Italian Conference on Algorithms and Complexity (CIAC 2000), March 2000. Springer LNCS vol. 1767, pp. 32-43.
- K. Y. Yeung, D. Haynor and W. L. Ruzzo, "Validating Clustering for Gene Expression Data," *Bioinformatics*, 17(4):309-318, 2001.
- A. D. Keller, M. Schummer, L. Hood and W. L. Ruzzo, "Bayesian Classification of DNA Array Expression Data." Univ. of Wash. Dept. of Comp. Sci. and Eng. Technical Report UW-CSE-2000-08-01, August, 2000.
- K. Y. Yeung, M T. Barrett, J. Delrow, P. L. Blount, L Hsu, W. L. Ruzzo, B. J. Reid, and P. S. Rabinovitch, "Expression analysis of Barrett's epithelium and normal gastrointestinal tissues," Technical Report UW-CSE-2000-11-01, Nov. 2000.
- K. Y. Yeung and W. L. Ruzzo, "An empirical study of Principal Component Analysis for clustering gene expression data." *Bioinformatics*, 2001 (to appear).
- K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering of gene expression data." Invited to appear in *Interface 2001: Frontiers in Data mining and Bioinformatics*, The 33rd Symposium on the Interface of Computing Science and Statistics, Costa Mesa, CA, June 2001. See also Technical Reports UW-CSE-2001-04-02; Statistics Department TR 396.

## Institutional and professional service in last five years:

- 95--96: Executive Committee, Lab Policy, Publications, Development (ch), UW Bothell Curric & planning

- 96--97: UW Bothell CSS Search Committee, TA Assignments, Conflict of Interest Oversight (ch), COE Educational Policy,
- 97--98: UW Bothell Liason, Community College Liason, 142/143 Coordination (ch), Conflict of Interest Oversight (ch), COE Educational Policy (ch), COE ad hoc ENGR Restructuring, University Grant and Contract Process Enhancement Team, University (Search) Advisory Committee on the Dean and Vice Provost of UW Bothell, UW Bothell CSS Search committee (twice).
- 98--99: Lab Policy (ch), UW Bothell Liason, COE Educational Policy (ch), University Grant and Contract Process Enhancement Team, and chair of the associated Project Implementation Team, University (Search) Advisory Committee on the Dean and Vice Provost of UW Bothell, UW Bothell CSS Search committee.
- 00--01: Curriculum, 142-143 Stewardship, ACMS Program Liason (ch), CMB Program Liason (ch)

**States in which registered:** none

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

---



# Salesin, David

*Professor (1992 Ass't, 1996 Assoc, 2001 Full)*

---

## **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. 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.
- Method and system for reproducing color images as duotones, with J. L. Power, B. S. West, and E. J. Stollnitz. Patent number 5,982,924, issued November 9, 1999.
- Method and system for generating graphic illustrations according to a stroke texture and a tone, with G. Winkenbach, M. Salisbury, S. Anderson, and R. Barzel. Patent number 5,847,712, issued December 8, 1998.
- Method and system for editing multiresolution images at fractional-levels of resolution using a wavelet representation, with D. Berman and J. Bartell. Patent number 5,666,475, issued September 9, 1997.
- Method and system for editing the general sweep and detail of a figure with a curve, with A. Finkelstein. Patent number 5,594,853, issued January 14, 1997.

## **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.
- 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. Duchamp, and W. Stuetzle. Proceedings of SIGGRAPH 2000, in Computer Graphics Proceedings, Annual Conference Series, July 2000. To appear.
- Environment matting extensions: toward higher accuracy and real-time capture, with Y.-Y. Chuang, D. Zongker, E. Hindorff, B. Curless, and R. Szeliski. Proceedings of SIGGRAPH 2000, in Computer Graphics Proceedings, Annual Conference Series, July 2000. To appear.
- Example-based hinting of TrueType fonts, with D. E. Zongker and G. Wade. Proceedings of SIGGRAPH 2000, in Computer Graphics Proceedings, Annual Conference Series, July 2000. To appear.

- 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.
- Synthesizing realistic facial expressions from photographs, with F. Pighin, J. Hecker, D. Lischinski, and R. Szeliski. Proceedings of SIGGRAPH 98, in Computer Graphics Proceedings, Annual Conference Series, 75-84, July 1998. A video accompanying the paper was screened at the SIGGRAPH 98 Computer Animation Festival, Animation Theater program, July 1998; appears in the SIGGRAPH 98 Video Review; and was selected as one of 25 works for Art Futura 98 in Seville and Madrid, Spain, October 1998.
- 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.
- Hierarchical image caching for accelerated walkthroughs of complex environments, with J. Shade, D. Lischinski, T. DeRose, and J. Snyder. Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series, 75-82, August 1996. Also available as Department of Computer Science and Engineering Technical Report TR 96-01-06, University of Washington, 1996.
- Interactive multiresolution surface viewing, with A. Certain, J. Popovic, T. DeRose, T. Duchamp, and W. Stuetzle. Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series, 91-98, August 1996. Also available as Department of Computer Science and Engineering Technical Report TR 96-01-07, University of Washington, 1996.
- The virtual cinematographer: a paradigm for automatic real-time camera control and directing, with L. He and M. Cohen. Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series, 217-224, August 1996.
- Comic chat, with D. Kurlander and T. Skelly. Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series, 225-236, August 1996.
- Reproducing color images as duotones, with J. L. Power, B. S. West, and E. J. Stollnitz. Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series, 237-248, August 1996. Also available as Department of Computer Science and Engineering Technical Report TR 96-01-08, University of Washington, 1996.
- Multiresolution video, with A. Finkelstein and C. Jacobs. Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series, 281-

- 290, August 1996. Also available as Department of Computer Science and Engineering Technical Report TR 96-01-01, University of Washington, 1996.
- 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.
  - Clustering for glossy global illumination, with P. H. Christensen, D. Lischinski, and E. J. Stollnitz. ACM Transactions on Graphics 16(1): 3-33, January 1997. An earlier version is available as Department of Computer Science and Engineering Technical Report Illumination TR 95-01-07, University of Washington, 1995.
  - Performance-driven hand-drawn animation, with I. Buck, A. Finkelstein, C. Jacobs, A. Klein, J. Seims, R. Szeliski, and K. Toyama. To appear in the Proceedings of the First Annual Conference on Non-Photorealistic Rendering and Animation (NPAR 2000, Annecy, France), 2000.
  - Resynthesizing facial animation through 3D model-based tracking, with F. Pighin and R. Szeliski. Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV '99, Kerkyra, Greece), 143-150, 1999.
  - Interactive arrangement of botanical L-system models, with J. L. Power, A. J. B. Brush, and P. Prusinkiewicz. Proceedings of the 1999 Symposium on Interactive 3D Graphics (Atlanta), 175-182, 1999.
  - Progressive previewing of ray-traced images using image plane discontinuity meshing, with F. P. Pighin and D. Lischinski. Proceedings of the Eighth Eurographics Workshop on Rendering (Saint-Etienne, France), 115-125, 1997.
  - Declarative camera control for automatic cinematography, with D. Christianson, S. E. Anderson, L. He, D. S. Weld, and M. F. Cohen. Proceedings of AAAI '96 (Portland, Oregon), 148-155, 1996. An earlier version appeared as Department of Computer Science and Engineering Technical Report TR 95-01-03, University of Washington, 1995.
  - A truly animated course. Paideia (newsletter of the Office of Undergraduate Education at the University of Washington) 5(2): 10-11, 1997.
  - Realistic facial animation using image-based 3D morphing, with F. Pighin, J. Auslander, D. Lischinski, and R. Szeliski. Department of Computer Science and Engineering Technical Report UW-CSE-97-01-03, University of Washington, 1997.

**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.
- National Science Foundation Presidential Faculty Fellow Award, 1995-98.
- Alfred P. Sloan Research Fellowship, 1995-97.
- Office of Naval Research Young Investigator Award, 1995-98.

- National Science Foundation Young Investigator Award, 1993-95.
- Stanford University Nominee for ACM Dissertation Award, 1991.
- AT&T Graduate Fellowship, 1988-91.
- Hertz Foundation Research Grant, 1987, 1988.
- Stanford University School of Engineering Fellowship, 1986-87.
- Winston Churchill Fellowship Winner (declined, in order to work at Lucasfilm), 1983.
- Brown University Prize for Outstanding Undergraduate Research, 1983.
- Elected to Sigma Xi, 1983.
- Elected to Phi Beta Kappa, junior year, 1982.
- National Merit Scholarship, 1979-83.
- New York State Regents Scholarship, 1979.
- Rensselaer Polytechnic Institute Mathematics and Science Award, 1978.
- Bausch and Lomb Science Award, 1978.

**Institutional and professional service in last five years:**

- 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.
- Raster Imaging & Digital Typography Program Committee, 1998.
- Computers and Graphics editorial advisory board member, 1992-98.
- ACM Symposium on Computational Geometry Video Review Committee, 1997.
- SIGGRAPH Technical Sketches Committee, 1996.
- Eurographics Technical Programme Committee, 1996.
- Eurographics Workshop on Rendering. Program committee member, 1993, 1995, 1996.
- National Science Foundation Institutional Infrastructure Site Visit Committee, 1995.
- Image Synthesis Conference at the 1995 IS&T/SPIE Symposium. Conference committee member, 1995.

**States in which registered:** none

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

---

# Seitz, Steven

*Assistant professor (2000 Ass't)*

---

## 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, October 1997 -- August 1998.

## Principal publications of last five years:

- K. N. Kutulakos and S. M. Seitz, A theory of shape by space carving, *International Journal of Computer Vision*, Marr Prize Special Issue, 2000, 38(3), pp. 199-218
- J. Gemmell, C. L. Zitnick, T. Kang, K. Toyama, and S. M. Seitz, Gaze-awareness for Videoconferencing: A Software Approach, *Multimedia*, to appear
- S. M. Seitz and C. R. Dyer, Photorealistic scene reconstruction by voxel coloring, *International Journal of Computer Vision*, vol 35, no. 2, 1999, pp. 151-173
- S. M. Seitz and C. R. Dyer, View-invariant analysis of cyclic motion, *International Journal of Computer Vision*, 1997, vol 25, no. 3, 1997, pp. 231-251
- 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*, to appear
- S. M. Seitz, A. Kalai, and H. Shum, Omnivergent stereo, *International Journal of Computer Vision*, accepted for publication, 2001
- S. M. Seitz and C. R. Dyer, Cyclic motion analysis using the period trace, *Motion-Based Recognition*, M. Shah and R. Jain, eds., Kluwer, Boston, 1997
- S. M. Seitz, The space of all stereo images, *Proc. International Conference on Computer Vision*, 2001, to appear.
- L. Zhang and S. M. Seitz, Image-based multiresolution shape recovery by surface deformation, *Proc. SPIE*, 2001 2000, pp. 209-218
- F. Dellaert, S. M. Seitz, C. E. Thorpe, S. Thrun, Feature correspondence: a Markov Chain Monte Carlo approach, *Proc. Neural Information and Processing Systems*, 2000, to appear
- J. Popovic, S. M. Seitz, M. Erdmann, Z. Popovic, and A. Witkin, Interactive manipulation of rigid body simulations, *Proc. SIGGRAPH*, 2000, pp. 209-218
- F. Dellaert, S. M. Seitz, C. E. Thorpe, S. Thrun, Structure from motion without correspondences, *Proc. Computer Vision and Pattern Recognition Conference*, 2000, pp. 557-564
- S. Vedula, S. Baker, S. Seitz, and T. Kanade, Shape and motion carving in 6D, *Proc. Computer Vision and Pattern Recognition Conference*, 2000, pp. 592-598
- S. B. Kang, S. Seitz, and P.-P Sloan, Visual tunnel analysis for camera planning and visibility prediction, *Proc. Computer Vision and Pattern Recognition Conference*, 2000, pp. 195-202
- H. Y. Shum, A. Kalai, and S. M. Seitz, Omnivergent stereo, *Proc. Seventh International Conference on Computer Vision*, 1999

- K. N. Kutulakos and S. M. Seitz, A theory of shape by space carving, Proc. Seventh International Conference on Computer Vision, 1999, pp. 307-314
- S. M. Seitz and P. Anandan, Implicit scene reconstruction from probability density functions, Proc. Computer Vision and Pattern Recognition Conference, 1999, pp. 28-34.
- S. M. Seitz and K. N. Kutulakos, Plenoptic image editing, Sixth International Conference on Computer Vision, 1998, pp. 17-24
- S. M. Seitz and C. R. Dyer, Photorealistic scene reconstruction by voxel coloring, Proc. Computer Vision and Pattern Recognition Conference, 1997, pp. 1067-1073
- S. M. Seitz and C. R. Dyer, Toward image-based scene representation using view morphing, Proc. 13th International Conference on Pattern Recognition, 1996, pp. 84-89
- S. M. Seitz and C. R. Dyer, View morphing, Proc. SIGGRAPH 96, 1996, pp. 21-30

**Scientific and professional societies of which a member:**

- IEEE

**Honors and awards:**

- NSF CAREER AWARD, May 2000
- DAVID MARR PRIZE, for the best paper at the International Conference on Computer Vision, September 1999
- BEST GRADUATE STUDENT RESEARCHER AWARD, U.W. Madison, May 1998

**Institutional and professional service in last five years:**

- University committees:  
Engineering Writing Assessment (University of Washington)  
Robotics Minor Committee (Carnegie Mellon University)  
Faculty Recruiting Committee (Carnegie Mellon University)
- EDITOR of The Visual Computer, International Journal of Computer Graphics, 2000-present
- GUEST EDITOR of ACM Computer Graphics Special Issue on Applications of Computer Graphics for Computer Vision, November, 1999, with Richard Szeliski
- CO-ORGANIZER, ACM SIGGRAPH Course on 3D Photography, 1999 and 2000
- CO-ORGANIZER, IEEE CVPR Course on 3D Photography, 1999
- PROGRAM COMMITTEES: ACM SIGGRAPH, 2000 and 2001; IEEE Computer Vision and Pattern Recognition Conference, 1998 and 2000; International Conference on 3D Digital Imaging and Modeling, 2001; Indian Conference on Computer Vision, Graphics, and Image Processing, 2000; Workshop on 3D Structure from Multiple Images of Large-scale Environments (SMILE), 2000; IEEE Workshop on Multi-View Modeling & Analysis of Visual Scenes, 1999; IEEE Work-shop on Photometric Modeling for Computer Vision and Graphics, 1999
- PANELIST, NSF Panel on Symbolic, Numeric, and Geometric Computing; NSF Panel on Robotics and Human Augmentation

**States in which registered:** none

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

# Sengupta, Rimli

*Research assistant professor (1999 Ass't)*

---

## Education:

- Ph.D., Computer Science, Georgia Tech, 1995
- M.S., Computer Science, Georgia Tech, 1993
- M. Tech., Electrical Engg., Indian Institute of Technology, Kanpur, India, 1989
- B. Engg., Electronics and Telecommunication Engg., Jadavpur University, Calcutta, India, 1987

## Other Experience:

- 8/95 - 8/99: Assistant Professor, Computer Science, Rose-Hulman Institute of Technology, Terre Haute, Indiana.
- 6/98 - 8/99: Visiting Scholar, Computer Science and Engg., University of Washington, Seattle.
- 6/97 - 8/97: Visiting Researcher, Computer Science, University of Texas, San Antonio.
- 9/94 - 3/95: Instructor, Georgia Tech.

## Principal publications of last five years:

- R. Sengupta and H. Venkateswaran  
"Non-cancellative Boolean Circuits: A Generalization of Monotone Boolean Circuits", *Theoretical Computer Science*, 237:1-2 (2000), 197-212.
- D. Mutchler, C. Anderson, C. Laxer, R. Sengupta, F. Young  
"CS1 Closed Laboratories = Multimedia Materials + Human Interaction".  
In Proc. 4th ACM SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, June 1999.
- S. R. Das, R. Castañeda, R. Sengupta, J. Yan  
"Comparative Performance Evaluation of Routing Protocols for Mobile Ad hoc Networks". In Proc. 7th International Conference on Computer Communication and Networks, IEEE Computer Society Press, October 1998.
- R. Sengupta and H. Venkateswaran  
"A Lower Bound for Monotone Arithmetic Circuits Computing 0-1 Permanent", *Theoretical Computer Science* 209:1-2 (1998), 389-398.
- R. Sengupta  
"Cancellation Is Exponentially Powerful for Computing the Determinant", *Information Processing Letters* 62 (1997), 177-181.
- R. Sengupta and D. Mutchler  
"Design vs. Use: Do We Have to Choose?"  
Position paper for NSF Workshop on Future Directions of CS2, Duke University, North Carolina, 1997.
- R. Sengupta and H. Venkateswaran  
"Non-cancellative Boolean Circuits: A Generalization of Monotone Boolean Circuits". In Proc. 16th Foundations of Software Technology and Theoretical Computer Science, *Lecture Notes in Computer Science*, 108 (1996), 298-309.

## Honors and awards:

- NSF POWRE award, 1998-99.
- Sloan/DOE Fellowship in Computational Biology, 1999-2001.
- Volunteer of the month (for voluntary services to the adult literacy program at the county public library in Terre Haute, Indiana) March 1998.

**Institutional and professional service in last five years:**

- Institutional and professional service while at Rose-Hulman (1995 - 1999):  
Member, Commission on Institutional Assessment (1996 - 1998)  
Co-organizer, Freshman laptop orientation (1996, 1997)  
Technical Communication faculty advisor  
Participant, Women in Math and Engg. Program (1996) designed to attract high school girls to Math, Science and Engg.  
President, Rose-Hulman chapter of Phi-Beta-Delta (1997 - 1998)

**States in which registered:** none

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

---



# Shapiro, Linda

*Professor (1986 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

## **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

**States in which registered:** none

**Level of activity:** prof 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 Effects of Communication 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 Impementation 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:** prof societies - medium; research - high; consulting/summer work in industry - low.

---

# Suciu, Dan

*Assistant professor (2000 Ass't)*

---

## 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:

- Serge Abiteboul, Peter Buneman, Dan Suciu, "Data on the Web: from Relations to Semistructred Data and XML", Morgan Kaufmann, 1999, ISBN 1-55860-622-X.
- Mary Fernandez, Dan Suciu, Wang-Chiew Tan, "SilkRoute: Trading between Relations and XML", 9th International World Wide Web Conference (WWW9), 2000, pages 723--746.
- Hartmut Liefke, Dan Suciu, "XMill: An Efficient Compressor for XML data", ACM SIGMOD Conference on Management of Data (SIGMOD), 2000, pages 153-164. Best paper award.
- Alin Deutsch, Mary Fernandez, Dan Suciu, "Storing semistructured data with STORED", ACM SIGMOD Conference on Management of Data (SIGMOD), 1999, pages 431-442.
- Alin Deutsch, Daniela Florescu, Mary Fernandez, Alon Levy, Dan Suciu, "A query language for XML", 8th International World Wide Web Conference (WWW8), 1999, pages 77-91.
- Peter Buneman, Mary Fernandez, Dan Suciu, "UnQL: A Query Language and Algebra for Semistructured Data Based on Structural Recursion", VLDB Journal, 9(1), 2000, pages 76-110.
- Tova Milo, Victor Vianu, Dan Suciu, "Type checking for XML transformations", ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), 2000, pages 11-22.
- Tova Milo, Dan Suciu, "Index structures for path expressions", International Conference on Database Theory (ICDT), 1999, pages 277-295.
- Mary Fernandez, Dan Suciu, "Optimizing Regular Path Expressions Using Graph Schemas", IEEE International Conference on Data Engineering, 1998, pages 14-23.

- Peter Buneman, Susan Davidson, Mary Fernandez, Dan Suciu, "Adding Structure to Unstructured Data", International Conference on Database Theory (ICDT), 1997, pages 336-350.

**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:**

- IEEE International Conference on Data Engineering (ICDE), 2001. (vice chair)
- International World Wide Web Conference (WWW10), 2001 (deputy vice chair)
- ACM SIGMOD International Conference on Management of Data (SIGMOD), 2001
- International Conference on Database Theory (ICDT), 1999 and 2001
- ACM International Conference on Information and Knowledge Management (CIKM), 2000
- Deductive and Object-oriented Databases (DOOD), 2000
- International Conference on Very Large Data Bases (VLDB), 2000
- Conference on Extending Database Technology (EDBT), 1999
- ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), 1998
- Dagstuhl Seminar on Semistructured Data, 2001 (Co-organizer)
- WebDB 2000 (Program Co-chair)
- DIMACS Workshop on Data Processing on the Web: A Look into the Future, 2000 (Co-organizer)
- Query Processing for Semistructured Data and Non-Standard Data Formats, 1999 (Program Co-chair)
- Workshop on Management of Semistructured Data (Program Co-chair)

**States in which registered:** none

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

---

# 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.:

- Scientific Advisor to Learner Empowerment, Inc. 1999-2000.

## Principal publications of last five years:

- Tanimoto, S.-L. and King, J., and Rice, R.-E. 2000.  
Learning mathematics with image processing: Constructing cylindrical anamorphoses. International Conference on Mathematics/Science Education and Technology, San Diego, CA, Feb. 5-8, 2000.
- Johnson, D.-B. and Tanimoto, S.-L. 1999.  
Reusing web documents in tutorials with the current-documents assumptions: Automatic validation of updates.  
Proc. ED-MEDIA 99, Seattle, WA, June, 1999.
- Bricker, L.-J., Baker Bennett, M.-J., Fujioka, E., and Tanimoto, S.-L. 1999.  
Colt: A system for developing software that supports synchronous collaborative activities.  
Proc. ED-MEDIA 99, Seattle, WA, June, 1999.
- Tanimoto, S.-L. 1998.  
Connecting middle school mathematics to computer vision and pattern recognition. Int'l J. of Pattern Recognition and Artificial Intelligence. Vol.~12, No.~8, pp.1053-1070.
- Tanimoto, S.-L. and Bernardelli, C.-E. 1998.  
The design and implementation of Vedo-Vedi, a visual language for human communication in the Internet. Technical Report, Dept of Computer Science and Engineering, Univ. of Wash., Seattle, April 1998.
- Tanimoto, S.-L. and Bernardelli, C.-E. 1998.  
Extensibility in a visual language for web-based interpersonal communication. Technical Report, Dept of Computer Science and Engineering, Univ. of Wash., Seattle, March 1998.
- Cinque, L., Lecca, F., Levialdi, S., and Tanimoto, S. 1998.  
Image retrieval using rich region descriptions.  
Proc. Int'l Conf. on Pattern Recognition, Brisbane, Australia.
- Tanimoto, S.-L. 1997.  
Representation and learnability in visual languages for web-based interpersonal communication (Keynote lecture).  
Proceedings of the International Symposium on Visual Languages, Capri, Italy, pp.2-10.
- Baker, M., Bricker, L., and Tanimoto, S.L. 1997.  
Cooperative interaction techniques for graphical objects in a collaborative activity. Technical report 97-04-03, Dept. of Computer Science and Engineering, Univ. of Washington, April.

- Allen, R., Cinque, L., Tanimoto, S.L., Shapiro, L.G., and Yasuda, D. 1997. A parallel algorithm for graph matching and its MasPar implementation. IEEE Trans.Parallel and Distributed Systems, Vol.8, No.5, pp.490-501.
- Cinque, L., Yasuda, D., Shapiro, L.G., Tanimoto, S.L., and Allen, R. 1996. An improved algorithm for relational distance graph matching. Pattern Recognition, Vol.29, No.2, pp.349-359.
- Jakobovits, R., Lewis, L.M., Ahrens, J.P., Shapiro, L.G., Tanimoto, S.L., and Brinkley, J.F. 1996. A visual database system for scientific research. J. Visual Languages and Computing, Vol.7, No.4, pp.361-375.
- Tanimoto, S.L. 1995. Exploring mathematics with image processing. Proceedings of the 1995 IFIP World Conference on Computers in Education, Birmingham, UK. London: Chapman & Hall, pp.805-814.
- Robinson, R.V., Cook, D., and Tanimoto, S.L. 1995. Programming agents using visual rules. Proceedings of the 1995 International Symposium on Visual Languages, Darmstadt, Germany, Sept. 5-9, pp.13-20.

**Scientific and professional societies of which a member:**

- IEEE Computer Society
- Association for Computing Machinery
- National Council of Teachers of Mathematics.

**Honors and awards:**

- IEEE Fellow.
- IEEE Computer Society Outstanding Contribution Award.

**Institutional and professional service in last five years:**

- Faculty Council on University Libraries, 1999-present.
- Search committee for Director of Libraries, 2000.
- International Association for Pattern Recognition, Education Committee Chair, 1998-2000.
- International Symposium on Visual Languages, Co-General Chair, 2000; Steering Committee, Program Committee.
- Program Co-Chair, International Conference on Pattern Recognition subconference on Algorithms and Techniques, 1998.
- NSF IGERT program at the University at Buffalo on Geographical Information Science: Advisory Board member.
- Program Committee Member for: Distributed Multimedia Systems 1997, 1998, Multimedia Modeling 1995, 1997, 1998, Discrete Geometry for Computer Imaging 1997. International Conference on Image Analysis and Processing 1999, 2001.

**States in which registered:** none

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

---

# **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:**

- "Minimal Adaptive Routing on the Mesh with Bounded Queue Size" (with D. D. Chinn and T. Leighton). *Journal of Parallel and Distributed Computing*, vol. 34, no. 2, May 1, 1996, 154-170.
- "Time-Space Tradeoffs for Undirected Graph Traversal by Graph Automata" (with P. Beame, A. Borodin, P. Raghavan, and W. L. Ruzzo). *Information and Computation*, vol. 130, no. 2, November 1, 1996, 101-129.
- "An Algorithm for Finding Novel Gapped Motifs in DNA Sequences" (with E. Rocke). *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, New York, NY, March 1998, 228-233.
- "A Time-Space Tradeoff for Undirected Graph Traversal by Walking Automata" (with P. Beame, A. Borodin, P. Raghavan, and W. L. Ruzzo). *SIAM Journal on Computing*, vol. 28, no. 3, 1999, 1051-1072.
- "A Linear Time Algorithm for Finding All Maximal Scoring Subsequences" (with W. L. Ruzzo), *Seventh International Conference on Intelligent Systems for Molecular Biology*, Heidelberg, Germany, August 1999, 234-241.
- "An Exact Method for Finding Short Motifs in Sequences, with Application to the Ribosome Binding Site Problem", *Seventh International Conference on Intelligent Systems for Molecular Biology*, Heidelberg, Germany, August 1999, 262-271.
- "An Exact Algorithm to Identify Motifs in Orthologous Sequences from Multiple Species" (with M. Blanchette and B. Schwikowski), *Eighth International Conference on Intelligent Systems for Molecular Biology*, San Diego, CA, August 2000, 37-45.
- "A Statistical Method for Finding Transcription Factor Binding Sites" (with S. Sinha), *Eighth International Conference on Intelligent Systems for Molecular Biology*, San Diego, CA, August 2000, 344-354.



- "Quality Control in Manufacturing Oligo Arrays: a Combinatorial Design Approach" (with R. Sengupta), Pacific Symposium on Biocomputing, Mauna Lani, Hawaii, January 2001, 348-359.
- "Finding Motifs Using Random Projections" (with J. Buhler), Proceedings of the Fifth Annual International Conference on Computational Molecular Biology, Montreal, Canada, April 2001, 67-74.
- "Equireplicate Balanced Binary Codes for Oligo Arrays" (with N. Alon, C. J. Colbourn, and A. C. H. Ling). Submitted for publication, 2001.
- "Construction of Optimal Quality Control for Oligo Arrays" (with C. J. Colbourn and A. C. H. Ling). Submitted for publication, 2001.
- "Identifying Functional Elements by Comparative DNA Sequence Analysis". Invited Insight/Outlook commentary, Genome Research, July 2001.

**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 chair, 37th Annual Symposium on Foundations of Computer Science, Burlington, Vermont, October 1996.
- Site visit committee member, National Institutes of Health proposed Center for Computational Biology and Bioinformatics, University of Pittsburgh, July 1999.
- Editor, Information and Computation, 1984-1998.
- University of Washington Computational and Mathematical Biology steering committee, 1999-present.
- Departmental service: Associate Chair, 1993-98; 142/143 coordination and community college liaison committee, 1998-99; Commercialization oversight committee, 1998-2000 (chair); Ph.D. review of progress, 1999-2000; Computational biology liaison, 1999-2000.

**States in which registered:** none

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

# 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:

- Founder, Nimble.com (1999)
- Founder, AdRelevance Inc. (1998)
- Consultant, Excite Inc. (1998)
- Founder, Netbot Inc. (1996) Creator of Jango shopping search

## 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:

- Levy, A. and Weld, D., "Intelligent Internet Systems" *Artificial Intelligence*, 118(1--2):1--14, April, 2000.
- Etzioni, O., Golden, K. and Weld, D., "Sound and Efficient Closed-World Reasoning for Planning," *Artificial Intelligence*, 89:113--148, 1997.
- Kushmerick, N., Hanks, S. and Weld, D., "An Algorithm for Probabilistic Planning," *Artificial Intelligence*, 76:239--286, July 1995.
- Hanks, S. and Weld, D., "A Domain-Independent Algorithm for Plan Adaptation," *Journal of AI Research*, 2:319--360, January 1995.
- Wolfman, S. and Weld, D. "The LPSAT Engine & its Application to Resource Planning," *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, August 1999.
- Smith, D. and Weld, D. "Temporal Planning with Mutual Exclusion Reasoning," *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, August 1999.
- Ives, Z. and Florescu, D. and Friedman, M. and Levy, A. and Weld, D. "An Adaptive Query Execution System for Data Integration," *1999 ACM Conference on Management of Data (SIGMOD-99)*, Philadelphia, PA, June 1999.
- Lau, T. and Weld, D., "Programming by Demonstration: An inductive learning formulation," *1999 ACM International Conference on Intelligent User Interfaces (IUI-99)*, Orlando, FL, January 1999.
- Weld, D. and Anderson, C. and Smith, D., "Extending Graphplan to Handle Uncertainty & Sensing Actions," *Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 9 pages, Madison, WI, July 1998.

- Smith, D. and Anderson, C. and Weld, D., "Contingent Graphplan," Fifteenth National Conference on Artificial Intelligence (AAAI-98), 9 pages, Madison, WI, July 1998.
- Anderson, C. and Smith, D. and Weld, D., "Conditional Effects in Graphplan," Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98), 9 pages, Pittsburg, PA, June 1998.
- Kushmerick, N. and Doorenbos, R. and Weld, D., "Wrapper Induction for Information Extraction," Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), 7 pages, Nagoya Japan, August 1997.
- Friedman, M. and Weld, D., "Efficient Execution of Information Gathering Plans," Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), 7 pages, Nagoya Japan, August 1997.
- Ernst, M. and Millstein, T. and Weld, D., "Automatic SAT-Compilation of Planning Problems," Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), 8 pages, Nagoya Japan, August 1997.
- Doorenbos, B., Etzioni, O. and Weld, D., "A Scalable Comparison-Shopping Agent for the WWW," First International Conference on Autonomous Agents (AGENTS-97), pages 39--48, Marina del Rey, CA, February 1997.
- Golden, K. and Weld, D., "Representing Sensing Actions: The Middle Ground Revisited," Fifth International Conference on Knowledge Representation and Reasoning (KR-96), pages 174--185, Cambridge, MA, November 1996.
- Wolfman, S. and Weld, D. "Combining Linear Programming and Satisfiability Solving for Resource Planning," Knowledge Engineering Review 15:1, 2000.
- Lau, T., Etzioni, O., and Weld, D. "Privacy Interfaces for Information Management," Communications of the ACM, October 1999.
- Anderson, C., Levy, A., and Weld, D. "Web-site management: the Tiramisu system" SIGMOD-99 Workshop on Databases and the Web, 1999.
- Weld, D., "Recent Advances in AI Planning," AI Magazine, 20:2, 93--123, Summer 1999.
- Perkowitz, M., Doorenbos, R., Etzioni, O. and Weld, D. "Learning to Understand Information on the Internet: An Example-Based Approach," Journal of Intelligent Information Systems, 8:133--153, 1997.
- Golden, K., Etzioni, O. and Weld, D., "Planning with Execution and Incomplete Information," University of Washington CSE Technical Report 96-01-09, April 1996.
- Weld, D., "COMPUTATIONAL TECHNIQUES: An Introduction to Programming in Mathematica," unpublished draft textbook, 121 pages, April 1995.

**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, 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
- John E. Bierwirth Scholarship, Yale University, 1979—82

- Wilfred Freeman Fellowship, Phillips Academy, 1978

**Institutional and professional service in last five years:**

- Advisory & Review Boards:  
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)
- Editorial Activities:  
Editorial Board Member, Artificial Intelligence (1999--)  
Guest Editor, Artificial Intelligence special issue on Intelligent Internet Systems (1998)  
Associate Editor, Journal of Artificial Intelligence Research (1993--1996)  
Guest Editor, Computational Intelligence special issue on qualitative reasoning (May 1992)  
Editorial Board of AI Communications (AICOM) declined
- Program Committee Chair:  
AAAI (1996)  
International Workshop on Qualitative Reasoning (1993)
- Program Committee Member  
AAAI (1990, 1991, 1992, 1993, 1996 (chair), 1997 (senior, declined)), 1999 (senior), 2000 (senior)  
AGENTS (1998, Area Chair for Software Agents)  
KR (1998 (declined))  
AIPS (1996 (declined), 1998 (declined), 1998 Planning Competition Rules Committee, 2000)  
International Workshop on Principles of Diagnosis (1998, declined)  
Pacific Rim International Conference of Artificial Intelligence, (1998, declined)  
European Conference on Planning, (1995, 1997)  
AAAI Workshop on Model-Based Reasoning (1990, 1991)  
International Workshop on Qualitative Reasoning (1987, 1989, 1992, 1993 (chair), 1994)  
Conference on Artificial Intelligence and Symbolic Mathematical Computation, Karlsruhe (1992)

**States in which registered:** none

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

---

# Wetherall, David

*Assistant professor (1999 Ass't)*

---

## 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

## Principal publications of last five years:

- Practical Network Support for IP Traceback,  
Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson,  
Proceedings of the 2000 ACM SIGCOMM Conference, Stockholm, Sweden, August 2000.
- A Content-based Technique for Eliminating Redundant Web Traffic  
Neil T. Spring; David Wetherall  
Proceedings of the 2000 ACM SIGCOMM Conference, Stockholm, Sweden, August 2000.
- Active network vision and reality: lessons from a capsule-based system  
David Wetherall  
17th ACM Symposium on Operating System Principles (SOSP'99), Kiawah Island, SC, December 1999.
- TCP Congestion Control with a Misbehaving Receiver  
Stefan Savage, Neal Cardwell, David Wetherall and Tom Anderson  
ACM Computer Communications Review, v 29, no 5, October 1999.
- Next Century Challenges: RadioActive Networks  
Vanu Bose, David Wetherall, and John Guttag  
ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'99), Seattle, WA, August 1999.
- ANTS: Network Services without the Red Tape  
David Wetherall, John Guttag and David Tennenhouse  
IEEE Computer, v 32, no. 4, pp 42-49, April 1999.
- Service Introduction in an Active Network  
David Wetherall  
Ph.D. Thesis, available as MIT/LCS/TR-773, February 1999.

## Scientific and professional societies of which a member:

- IEEE
- ACM

## Institutional and professional service in last five years:

- Program Co-Chair of IEEE OpenArch, 2001.
- Graduate Admissions Committee, 2000.
- Member of the SIGCOMM Program Committee, 2000.
- Member of the SIGCOMM Program Committee, 1999.
- Member of the 1st IEEE Workshop on Internet Applications Program Committee, 1999.

**States in which registered:** none

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

---

# 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:

- Derek Eager, Mary Vernon, and John Zahorjan, "Minimizing Bandwidth Requirements for On-Demand Data Delivery", to appear in IEEE Trans. on Knowledge and Data Engineering, Special Section of invited papers from MIS'99, Sep. 2001.
- Thu D. Nguyen, Christopher Peery, and John Zahorjan. "DDDDRRaW: A Prototype Distributed 3D Real-Time Rendering Toolkit for Commodity Clusters", Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS), May 2001.
- Thu Nguyen and John Zahorjan. "Image Layer Decomposition for Distributed Rendering on NOWs", Proc. 2000 International Parallel and Distributed Processing Symposium, Cancun, Mexico, May 1-5, 2000.
- Derek Eager, Mary Vernon, and John Zahorjan, "Bandwidth Skimming: A Technique for Cost-Effective Video-on-Demand", Multimedia Computing and Networking 2000 (MMCN00), San Jose, CA, Jan. 25-27, 2000.
- Derek Eager, Mary Vernon, and John Zahorjan, "Optimal and Efficient Merging Schedules for Video-on-Demand Servers", Proc. 7th ACM Multimedia Conf. (Multimedia '99), Orlando, FL, Oct. 30 - Nov. 5, 1999.
- Ruth E. Anderson, Thu D. Nguyen, and John Zahorjan, Cascaded Execution: Speeding Up Unparallelized Execution on Shared-Memory Multiprocessors. Proc. International Parallel Processing Symposium, May 1999.
- Thu Nguyen and John Zahorjan. Scheduling Policies to Support Distributed 3D Multimedia Applications. ACM SIGMETRICS / IFIP WG 7.3 Performance '98 Conf. on Meas. and Modelling of Comp. Sys., June 1998. (An expanded version of this paper is available as Technical Report UW-CSE-97-11-03, University of Washington, November 1997. (<http://www.cs.washington.edu/homes/thu/papers/rt-lb-policies.ps>.)
- S.G. Srinivasan, I. Ashok, Hannes Jonsson, Gretchen Kalonji, John Zahorjan. Parallel Short-Range Molecular Dynamics using the Adhara Runtime System. Computer Physics Communications Vol. 102 (no.1-3), pp. 28-43 (1997).
- S.G. Srinivasan, I. Ashok, Hannes Jonsson, Gretchen Kalonji, John Zahorjan. Dynamic-Domain-Decomposition Parallel Molecular Dynamics. Computer Physics Communications Vol. 102 (no.1-3), pp. 44-58 (1997).
- T. Nguyen, R. Vaswani, and J. Zahorjan. Maximizing Speedup through Self-Tuning of Processor Allocation. Proc. International Parallel Processing Symposium, May 1996.
- T. Nguyen, R. Vaswani, and J. Zahorjan. Parallel Application Characterization for Multiprocessor Scheduling Policy Design. In Job Scheduling Strategies for Parallel Processing, D.G. Feitelson and L. Rudolph (eds.), Springer-Verlag, Lecture Notes in Computer Science (1996).
- T. Nguyen, R. Vaswani, and J. Zahorjan. Using Runtime Measured Workload Characteristics in Parallel Processor Scheduling. In Job Scheduling Strategies for Parallel Processing, D.G. Feitelson and L. Rudolph (eds.), Springer-Verlag, 1996. Lecture Notes in Computer Science (1996).

**Institutional and professional service in last five years:**

- Program Committee Member, 18th ACM Symposium on Operating Systems Principles (2001).
- NSF Grant Review Panels, 2000
- Member of Board, Cleveland InfoTech Infusion Academy, 2000-.
- Webmaster, ACM SIGMETRICS, 1999-
- General Chair, 1997 ACM SIGMETRICS Conference on Measurement and Modelling of Computer Systems.
- Technical Vice-Chair, Performance of Distributed Systems, 17th International Conference on Distributed Computing Systems (1997).
- Program Committee Member, 2nd Symposium on Operating Systems Design and Implementation, October 1996.
- Program Committee Member, Workshop on Scheduling Strategies for Parallel Computing, May 1996.
- General Chair, 8th IEEE Symposium on Parallel and Distributed Processing, October 1996.

**States in which registered:** none

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

---

# Appendix I

(continued)

## D. Student exit survey comments for the past two years

Below are collected comments from our exit survey questions: **Do you have comments or suggestions to the department to better serve our students?** and **What courses would you like to see offered in the CSE department?**.

All comments are carefully reviewed each quarter by the Undergraduate Advising Unit and the Undergraduate Faculty Advisor. Exiting students have a particularly valuable perspective on the program because they have been through it all and have had the chance to evaluate for themselves how it fits together. In addition, a large percentage of these students have also worked in industry (approx. 50% and rising) and have gained an even larger point of view. Suggestions are evaluated and either acted upon directly or forwarded to the Curriculum Committee for further discussion and resolution. Of course, not all suggestions are implementable or may run counter to the Department's goals. However, these issues are brought up at subsequent student lunches so that all aware of the rationale for implementing or not implementing them.

- The quarter system sucks.
- I might suggest to add more flexibility offered to students in the computer engineering program, but I feel like the new program offers a lot more of that. Overall I've been extremely impressed with the program: great professors, interesting classes, smart & nice students, and lots of opportunities to get involved in research!
- **Unix course of how to operate and manage the unix system for new entries should be helpful (ACM Unix tutorial) 1999 graduate**
- More hardware design courses. (1999 grad)
- yes. Make students who wish to take cse457, take a computational geometry course first, so that one can better handle the math in graphics. If possible, increase the amount of disk space for CS accounts. Do not change the number of credits of a course during the middle of the quarter. This causes a myriad of annoying events, which will easily be overlooked by the busy CS student. Raise the mathematical standards of the incoming CS students. Perhaps, less chemistry prerequisites as a result. (1999)
- I would like to see animation offered more often. (1999 grad, so since have added more courses)
- EE 332 should not be a mandatory class for Computer Engineering students. EE 332 is an analog class. There should be a few more hardware design classes and a few less theory/programming classes for computer engineers. (1999)
- Department should try to keep the planning schedule as best as it can so students can easily plan for their schedule every quarter. 2. Core classes should be offered at least twice a year. This helps students don't have to wait a year or more to take one class for graduation. This problem will delay students' graduation. For instance, CSE471 is offered only in Autumn and is delayed to Winter this year. Many students have to take substitute classes for graduation, and they will miss the chance to learn about computer design. (We do need better course planning ...) 1999



- **Keep CSE451 using Unix. Have more X terminals. Keep using SGI machines for at least some Graphics course to get students familiar with more than one OS. Organize some sort of Alumni CS Dept club so that students can network with one another. Also perhaps connect these Alumni groups with Business-related entrepreneurial groups to encourage commercial tech-related ventures post-graduation. This would elevate the University's profile and allow a synergy similar to that of Stanford and Silicon Valley. 1999**
- Lab space is tight and not all that well maintained. Information isn't distributed well.... i.e. cs-ugrads does NOT get to ce students (mailing lists corrected to include both student groups in cs-ugrads) 1999
- Hmm. The department at a whole is excellent. I really like the push to diversify it in the recent years. I would just plan more social activities for undergrads to get together.
- More space. I know you all know this, but... Dedicated labs for classes would be fabulous. In 468 last quarter having a dedicated lab was great! The software never got screwed up!
- I wanted to take a language, but because I was in the engineering school, and the first year doesn't count, I really couldn't. That was a bit disappointing.
- I think some more classes should be made prerequisites for others. 370 should be required before taking 378, and 421 a prerequisite for 431. I think this sequence makes sense and would keep the instructor from spending time bringing part of the class up to speed on topics other students have already covered. I'm sure there are some other courses that would benefit from sequencing. I think the length of a quarter is a real disadvantage, especially for outer core courses with a large project component. If some of these courses could run over two quarters, then more significant projects could be tackled. I found it a little frustrating that once a project really got going the quarter would be over. Perhaps some of the capstone courses could be extended, if not outer core courses. I wish some of the outer core and capstone courses would be offered more frequently. There were some courses I wanted to take, but couldn't because they just didn't fit into my schedule or sometimes a course would get bumped to a different quarter. I think most cs students spend more time in the department than I have, so this may not be an issue with the majority of students. The past two quarters Prof. Borning has taught the Computers, Ethics, and Society course. While I did not take this course, I think it should be made a requirement. It was my impression that many of my peers do not give these important issues enough thought and they very rarely came up in cse courses I took. (2000 graduate)
- An honors introductory programming course that fulfills a university science requirement. You could encapsulate 142,143,326,and 341 into a three quarter long sequence. Not only would this free up a quarter for accelerated students, but attract top university honor students to the CSE department. (2000)
- When I arrived at the University in Fall 1995, I hadn't used a computer for anything more than word processing and gaming (Atari and Nintendo). Now, five years later, I have a firm grasp on what computers are, what they can do, and the major issues in efficient development and improvement of them. Truthfully, I cannot think of a computer science project where, maybe after some necessary introduction, I could not impressively contribute. Thanks. (2000)
- Without this research (Graphics), I wouldn't have gotten into grad school. I seriously urge every student who's considering grad school to do what I did as soon as possible. (2000)

- The classes provide pretty good coverage so far. Maybe a computer system's design course (eg, motherboards). 468 and 471 provided a good background on how a processor works, but there was no class that showed how the processor interfaces with the rest of the motherboard. (2000)
- It is a fundamental error on the part of the CSE department to not offer a course in the Ethics of Research and Development in Computing Technologies. A course should also be offered at the undergraduate level for HCI.
- Career counselor would be nice. Current academic counselors are awesome.
- It would be better if the CSE dept. would teach their own EE courses. Especially now that they are essentially only 2 required. They could gear them better toward computer applications.
- I was a little disappointed that my distribution of classes did not cover the 8 credits of Engr., Math, Science Electives. I think that one needs to be explained a little bit better; and also set apart from the Comp E. Electives better. 451, 467, and 477 were INCREDIBLE classes. I learned so incredibly much from them, and like my earlier response, I loved the hands-on nature of the courses. CSE471, while informative, only held my interest on the project half of class. The Entire final was not based at all on the project, but instead on only the lecture material f the last few weeks. So I was a little disappointed.  
The EE half of things was also very informative. I really loved to learn everything that 215, 233, 321, and 322 had to offer. However, the excessive lab-work, and the concentration of accuracy over learning really distracted me, since I feel that my University experience should be their for the learning, not as much for the applied. (The industry is so diferent, regardless of all the "group activities" that classes attempt.)
- 2 Quarter Course -- Software engineering joint some other department (eg. Business). 10 weeks just doesnt seem like enough time to get through the complete development cycle.
- There should be at least one class which focuses on real world programming. IE - how things work in the real world -- and I don't mean cse 403.
- security; a REAL computation linguistics class, not Ling 472.
- more web/internet courses (have since added Internet capstone)
- I think the courses currently offered are fine. However, I would like to see less cancelations...they are really annoying.
- More classes that invovle interaction between CS and other fields - such as a biology or human research program or class.
- Oh, I 'unno. Being third of seven co-authors on a paper published at this year's SIGGraph is kinda sweet. :)  
Now if I could only get the UW to pay for me to go... :)  
Seriously, research is interesting. I learned I'm a better software engineer than researcher, but it's been very educational all the same.  
Being on a very loose schedule without any sort of "credits" or formal plan has been a boon as well, since I've tended to get lost in a lot of other projects as well.
- Don't expand. ^\_^;  
Expansion baaaad. Staying small lets one stay elite. Expanding large means less quality education for people, and lower caliber students to get that education.

- Java programming
- I would like to see more networking classes. I would also like to see more in depth studies on the I/O subsystem of an Operating System.
- Provide more computers. Expanding the department has put a crunch on the number of computers available in the prime time hours. It is not always possible to rework your schedule to avoid the busy hours.
- Better courses on Networks, especially as it relates to the Internet.
- Get rid of NACHOS for CSE 451-- I would have much rather started at the ground and learned the basics in a coding environment, while learning about advanced topics in a higher level.
- A course on Windows Programming / MFC. It seems like a lot of courses tried to stay away from Windows-centric programming, which is a real shame considering the majority of companies out there are Windows shops.
- I really wanted to take a robotics class, but the EE class always conflicted with my other (required) CS classes. I think there is enough of an interest to cover robotics in a CS/CE class (or have better synchronization between time schedules).
- Break up the senior design capstone class (cse 477) so that there aren't any labs in 477- they are covered in the pre-req classes. 10 weeks isn't enough time to do a project, let alone doing labs as well.
- I think this has gotten better, but when I took cse 341, we studied languages that I will probably never see again. Now they are talking about other languages that I have used in my internship (Perl, XML) that would have been great to have exposure to in 341. So, I guess the lesson is to make sure that the content of the classes is up to date.
- I was a comp-E major for 2 years. I would only recommend the program to somebody who really, REALLY, likes hardware and bad EE professors. I would almost suggest working a deal with the various departments to have people only apply for CS, and, if they enjoy 370/378 and EE 215 then to apply for the CE degree program.
- How about courses that are 5 credits instead of 3? These courses and the amount of homework for even the basic 3 credit courses puts the average "hard" 5 credit course to shame from most other departments in this school. People basically kill themselves in this department taking 4 or 5 classes just to make full time.
- More crossover classes with different fields (or at least advertise them more)
- I wish I would have known EE332 would not be required back in September of 1999. I believe, better communication with students is needed.
- I really didn't enjoy many of the inner core computer science classes. They were too focused on theory and mathematics. Also the pace was way faster than I felt was reasonable for a solid understanding of all of the interesting concepts.
- I was denied admission to both of the upper level CS classes that were the pinnacle of my focuses. I was really excited about these two capstone classes, and had worked up all of the pre-recs for. I would suggest that you send people interested in graphics or games to another university.

- Natural Language Processing - a practical rather than the survey course currently offered through the Ling Department.  
System Management - a practical course that would allow students to experiment with setting up networks, user profiles, system settings, scripting etc.
- I really feel like I did not get enough practical and coding experience while in the department. The theoretical foundation was very good but the projects were too "well" guided to be long-term beneficial. I guess I would like to see more hands-off projects that allow students to experiment more
- a more advanced graphics course
- My research experience was much better than the internship one.  
I worked on two truly exciting projects, and had a chance to actually design specific components of these projects. Professors' help and support was extraordinary.
- Career counselor would be helpful. Other than that you guys KICK ASS!!!
- Embedded systems, but it looks like that's what's happening now.
- Could have used some web programming in inner core, in preparation for outer core.  
Software Engineering with David Notkin was a super class.
- Add some web programming to inner core
- I would like to see greater coordination between the mathematics and computer science departments. The ACMS program is not yet an acceptable substitute, as there are a great many repetitive and wasted classes when taking in conjunction with a CS degree
- Move away from drill-oriented curriculum. I dare say that those for whom Computer Science is new are unprepared to promote the proper goals of computing.
- a second course in data structures. (at the 400 level)
- I really \*really\* like the attempts made here to be as language and platform neutral as possible. Instead of teaching a technology that is current now and will be out of date in three years, we learn the theories and ideas that are timeless. This more than anything is what is valuable about the cse degree. I can pick up a book and learn a new language or technology, but the skills I have gained in the cse dept here are much deeper and more fundamental than that.
- Classes inside the department were noticeably better than those outside. (with the exception of VLPA courses perhaps.) Physics would get a 5 if it weren't for the first year series I took. (Honors Physics - normal physics would have been better if it meant getting a better professor!) Math would get a 5 too if it weren't for Math 324 which I just didn't get along with. (Why wasn't hw collected again?) CSE 341 is a weak point IMHO.
- Everything here is great expect the EE classes. Especially EE 215. The professor involved had no business teaching that course. EE 233 was noticeably better (at least the professor had a clue) but still could have benefited from significant reorganization.
- The biggest problem with the current curriculum is a lack of structure. What I mean here is that there isn't enough of a progression in classes to allow for good buildup of the material. I can't tell

you how many times I've seen a professor look around and say "How many people have taken xxx" and half the class raises their hands. In that case, the prof usually caters to the most general audience, which means that fewer topics can be discussed. I think some of this could be alleviated if CSE took some ideas from other departments, which offer series of courses. (see Math, Physics, Music...) That way, at the end of the series, the professor can rely on what was taught all year long. One idea would be to join up 378, Compilers, and OS together into a year-long series. I think the progression from assembly language to compilers is very natural, and the concepts learned in the first quarter of 378 would come back in a very useful way to make for a solid OS foundation in the third quarter. At the end of that series, students would know how the whole thing works from the ground up. I think that would be a very satisfying experience. Personally, I feel those three should be required classes anyway...

Another series could be theoretical. 322, 326, 421. Again, my personal belief is that these should be mandatory. In my case, I took 326 from Professor Tampa. Needless to say, we covered about half the topics of 421! That's really another part of the problem- professors teach the classes differently. That's nice because it gives variety, and lets the professor tailor the class to his/her strong points, but it makes for class interoperability problems. A case in point is Dijkstra's Algorithm. I think I got a thorough introduction to this in no less than 4 classes! It's a nice algo and all, but it got old after a while. That's mostly because professors thought it was cool, so if they had some topic that was close-by, they'd take a little detour. And I'm sure that some people in each one of those four classes were seeing it for the first time! If there were a series, it would only be shown in detail once. Again, I think the similarities in material would allow for some creative teaching opportunities in the series, allowing more topics to be covered, or in more detail, in the same amount of time.

Series only really make sense for these sorts of core classes. I don't see Networks fitting into a series too well, but then it's also relatively well contained. Perhaps two series would be enough. The other thing to think about would be the hardware side. 370/467 really should go together. My 467 professor spent a couple weeks just brushing up on 370 stuff! Also note that the series doesn't HAVE to be taught by the same professor. Someone else could take over in the middle, as long as there was a clear understanding of what was covered in the previous lectures. It may be beneficial to have the later professors attend the earlier classes. (this not only lets the professor become acquainted with the class, but also track their progress very closely.)

My other rant is that some classes really should be given a few more credits. Most notably CSE 490ca. In reality, this class is like a very intense full-time job. It's hard to sign up and only take 5 credits!

The largest waste of my time at the U was the technical communications courses. I honestly believe they have reduced my ability to write and speak effectively. EE was painful, but at least in the end it helped me understand something new. I think the idea to have a technical communications course is a great one- those are very important skills which most people could use a little help with (myself included.) I'd recommend having the class moved into the department, though, as the teachers who teach it now have absolutely no clue what they are doing.

Anyway, take everything I've just said with a grain of salt. I had a great time in this department, and I learned so very much from many wonderful professors. I really feel we have a solid program, and hope to see it continue to improve!

- I think that getting more exposure to the students about the research available in the department will be very helpful because they can apply what they get in the classes to what technology exists in the industry.
- Drop more EE requirements for Comp. E. Allow current Comp E students to switch tracks so they

don't have to suffer what I suffered.

- 143 did not prepare me for the programming I needed to do in the department! I had a terrible time in 326 because of this. There should be an additional possibly optional course for people who don't have a programming background to take after 143 that would bridge the gap between 143 and the departmental classes. I know I didn't get enough from 143 because I got a 3.9 in it, but then couldn't do the programming required for the homework in 326. Also, people should be encouraged to do a co-op earlier (maybe right after 143) so they have some more software development experience before taking departmental classes.
- Have faculty advisors and possibly graduate student advisors. I needed more advice from faculty that the advisors couldn't really provide since they aren't professors! The advisors were great for what they provide, I just needed more monitoring from someone in the field. Also, career advising just for our field! We are in a unique situation. It is frustrating not to have anyone to talk to about our job market, interview styles, and job offers. The people in career services are of only limited help because they don't specialize in our field.
- Computer Networks, Software Engineering, Embedded Systems (which are already offered at the undergraduate level), but i'm not sure at the graduate level. When I do my master's I would like to specialize in these areas.
- I LOVED the CSE department -- all courses and professors included. I like it just the way it is. Perhaps the only improvement would be to allow more research involvement opportunities at the undergraduate level.
- Human-Computer Interaction  
Windows programming (other schools do it, like Cornell)  
Advanced C++ and Java programming (Stanford does it)  
Classes on game logic (CMU has one)
- Should be more opportunities for tutoring and mentoring
- I would like to see more courses about real world application of CS. For example, I took cse590ip, software entrepreneurship, and I found it very educational.
- Having CSE461 offered jointly with EE dept. does a great disservice to both the CSE people and the EE people.  
The CSE people are disappointed because the EE people want it easier with less programming assignments because they don't know how to very well.  
And the EE people think it is too hard and irrelevant for them to be doing programming assignments as EE majors.  
Both sides have a good point, and it really makes the class suck for half the class, and which half gets ripped off obviously has to do with which department is teaching it.  
I know a lot of people want to take it when their respective department teaches it, but it is hard to do that when STAR only lets half of the class be that major. **(pre-requisites adjusted twice in last two years)**
- Having most of the CS-oriented 400-level classes as 3 credits is hard, because:  
(a) you do way more work than a 3 credit class coupled with  
(b) it makes it incredibly hard to put together a schedule... if I take 2 400-level CSE classes.. 1 5 credit "other" class.. that's only 11 credits and still not full time!  
.. and taking 4 classes with a schedule like that is terrible, since there are pretty much no 1-2 credit classes.  
These two are a frustrating combination often times.

And now something positive:

With all the noise on campus about tolerance of diversity, I think everyone in CSE is doing great. I've never felt treated badly/different because of my gender, and the atmosphere has been nothing but positive. Thanks!

- For the senior electives, I liked how there was flexibility so you could choose electives that were interesting to you. The inner and outer classes were all really good. All the professors were awesome.
- Networks should be an outer core course (**done in spring 2001**)
- Also, 466 is a class with good intentions that doesn't serve anyone well, they offer it to comp sci, and comp e. but then try to teach about single transistor amplifiers etc. this is frustrating to comp e. students who have spent several quarters studying transistor amplifiers etc. and bewildering to comp. sci students who have never seen transistors at all, and don't know even basic things about electronics. It should either really be software, or really be hardware, and if it's hardware, it should have pre-reqs. The department overall does a good job, there are a lot of opportunities, and more every year, I think that in the 4 years since I have been here it has become enormously easier to get involved with research, and they are doing a better job of informing people of all the opportunities that exist (they could still do better in the future, advertising interesting courses like DSP classes etc. over the mailing list, like the college honors program does). When I first got here I couldn't find any research to get involved with, even at the lowest level, now it's easy.
- The department has admitted far too many students without expanding class schedule and size. I was not seriously affected, but frustration is growing very high among those graduating after me.  
Overall my education was great. We have a world-class faculty, and by far most of them are genuinely interested in undergraduate education.
- Encourage more women to apply ... also have more female teachers, are there any?
- More hardware classes building up to the capstone courses. More than one VLSI class
- Have more classes. Have more blending with Business. SBE is a great thing.
- I'd like to see Internet Systems or something similar become a real course instead of just a Special Topics class.
- More pure software engineering courses (2001 grad)
- I think EE courses targeted at CSE students would be much better and more effective for Computer Engineering majors. Many topics are not covered in a way that relates to the future work of CSE majors. Also, some topics, such as powerlines, are covered that have little or no application to CSE majors.  
I am glad to see that a CSE-specific Stat 390 class has been created. I really wish I had taken that instead of normal Stat 390.
- I think we definitely need more lab and meeting space. It gets difficult to do any work towards the end of the quarter when things start getting crowded.