SELF-STUDY REPORT

Department of Computer Science

University of Maryland at College Park

Department Web Site: http://www.cs.umd.edu/

This report is a summary of our Department's research, education program, and infrastructure – administration, space and budgets. It provides an assessment of the Department's current strengths and weaknesses, and suggests plans for the continuing advancement of the quality of our programs.

TABLE OF CONTENTS

graduate student population and our graduate education program; and discusses graduate student placement – an area of high priority in which we have made significant improvement but one in which more needs to be accomplished.

Infrastructure – Contains a summary of the Department's budget history and how it has dealt with recent budget cuts, overviews the organization of the administrative staff and describes activities that cannot be undertaken because of staff shortages, and describes the space available to the Department, and the severe challenges it faces to support current activities and grow new ones due to space limitations.

Appendices

Overview

Plan of Organization

Past Reviews

Faculty Research

Department Research Accounts

UMIACS Research Accounts

Private Sector Funding

Faculty Profile

Faculty Honors

Affiliated Institutes and Laboratories

2003 ITR Proposals Submitted

Undergraduate Program

Contents

Overview, Goals and Recent History

The Department of Computer Science at the University of Maryland is one of the oldest in the nation. We have been granting Ph. D. degrees for nearly 30 years and undergraduate degrees for 25. The Department currently has 43 full time tenure track faculty conducting research in most areas of the field. The tenure track faculty are supported by ten full time instructors, an administrative support staff of 15.5 and a technical support staff of six full time professionals. There are approximately 250 full time graduate students, mostly pursuing Ph. D.'s, and 1700 undergraduate majors. Our major goals (and associated challenges) are to:

Improve upon our national rating as a research department. Our department has been consistently rated 11-13 by surveys in U.S. News and World Report and the National Academy of Sciences. The faculty is very strong; every junior faculty member we have hired in the past fifteen years has won an NSF CAREER award. Many senior faculty are members or founding members of major technical societies (IEEE, ACM AAI, etc.) and have been honored for their achievements in many ways. Funding from all sources-especially NSF-is at all time highs. We have also been exceptionally successful with our submissions to NSF's ITR program. Our primary goal is to move the Department into the top 10 nationally. But the competition is fierce. **Challenges:**

- **Place our best graduate students** in better academic positions. Our record here has been improving, with recent graduates currently in tenured and tenure track positions at Brown, Rutgers and Toronto.
- Identify **resources, especially, adequate laboratory and office space**, that will enable the Department to move into strategically important areas of research in a very competitive national environment.
- Improve the services offered by our graduate office, which is currently significantly understaffed and operates using many of the same processes that were in place in the early 1980's.
- Improve our **collaboration** with the architecture, systems and robotics faculty in the **Electrical and Computer Engineering Department** to develop strong research programs in embedded computing and robotics.

Enrich our undergraduate education program by expanding the research opportunities for talented undergraduates, expanding and improving our project-oriented courses and making more extensive use of local industry and government laboratories in delivering undergraduate education. The Department is embarking on a major renovation of its undergraduate education program, redesigning many of the courses and bringing tenure track faculty into the classroom for all of the introductory courses. **Challenges:**

- **"Customer support" for undergraduates**. Our undergraduate office is understaffed, with inadequate manpower for advising, tutoring and working with students on extracurricular activities
- **Space for undergraduate activities**. The Department has no undergraduate lounge, no space to develop new laboratories for undergraduate education and research, and very limited space for teaching assistants to meet with students and help them with projects and technical questions. For our 1700 undergraduate students we have less than 300 square feet assigned to this function, and it is not uncommon to find students camped out in hallways waiting for their (brief) turn for assistance.

Develop far better relations with our "communities." Our Department has not done a good job of

maintaining ties with its alumni or with local and national industry (although many individual faculty do very well in establishing research relationships with industry). It has not had an active advisory board in many years. This is in large measure due to severe staff shortages in the chair's office, aggravated in the past year by budget rescissions that have resulted in losing two important staff positions. **Challenges**:

- Address the **staff shortage** by adding at least two new staff positions to the chair's office – one for public relations, alumni relations and fund raising, and a second as an administrative assistant to the chair.
- **Reconstitute the Department's Advisory Board** as a first step towards developing new ties to industry and government. We must make better use of both senior management in local industry and government, as well as some of our more successful alumni in industry, academia and government.
- Identify and **track our most successful alumni** – both as a source of potential gifts to support department programs, and to document the success of our graduates at all levels.

The remaining sections of this report on Research, Undergraduate Education and Graduate Education discuss these challenges at greater length in the context of the Department's overall activities and strengths.

Results of Previous Review

It has been over ten years since the Department was last reviewed. Both the internal and external reviews are available at Past Reviews. Here, we list the most critical weaknesses identified by the last external review committee, and provide our own assessment of whether these issues have been adequately addressed during the intervening decade.

1. **Lack of consensus and ineffective governing structure**. This was a serious problem at the time of the last review, which we believe has been largely solved, although without having to make any significant changes to the governing structure of the Department. In part, the solution was forced upon us several years ago when the campus required each Department to construct five-year strategic plans that identified the most important areas for growth, and plans for moving into those areas. In conjunction with UMIACS, the Department initially identified thrust areas of computer systems and software engineering and programming languages and recruited vigorously and successfully in these areas. While current resources for recruiting are severely limited by the overall financial condition of the State University system, there is consensus that we should build a computationally biology group (and associated education programs at both the graduate and undergraduate level). The review committee also identified the structure of the graduate program as another example of failure to achieve consensus (every faculty member had some complaint about it, but the structure persisted). Since that time we have completely re-organized the graduate program, and are in the process of doing so again this year.

2. **Excessive dependence on counting in evaluations, as opposed to "impact."** The external review identified reluctance on the part of the faculty to evaluate research on the basis of long-term impact. We do not believe that this is the case today. The Department's promotion process places especially high emphasis on external letters of recommendation, and we strongly urge our references to comment on the impact of the research conducted by our faculty. The importance of publications in highly refereed conferences is appreciated both at the Department and higher levels of our campus during the promotion process.

3. **Slow to move into new areas**. Prior to the 1992 review, and for several years afterwards, the Department was essentially unwilling to offer positions to senior candidates, clearly the best way to move into new areas. This was mostly a reluctance to stray from the strategy that had worked in the past—hiring talented junior

people and letting them "grow" on the job. The situation is now completely reversed. We did hire Joel Saltz as an Associate Professor shortly after the review (and this did allow us to move quickly and effectively into the high performance computing area. This is an area in which the Department is now strong, so that we have been able to maintain our strength even with Joel's recent departure for Ohio State), and we have also made several senior offers to candidates in graphics, theory, scientific computing and computer vision. Within the past three years we have hired a "senior" Assistant Professor with a short time frame for consideration for tenure (Iftode), one untenured Associate Professor (Varshney-who was quickly tenured), two tenured Associate Professors (Srinivasan and Jacobs), and one Full Professor (DeFloriani).

4. **Dependence on UMIACS**. The external review committee felt that the Department was not able to control its budget because of its dependence on UMIACS. The current chair was then UMIACS director. There were sources of tension between the Department and the Institute, mostly related to issues of permanent appointments and teaching loads for Computer Science faculty who did not have appointments in research institutes. Moreover, UMIACS was a relatively new entity at that time and was given more flexibility in budgetary decisions than the Department. These problems no longer exist. During the past few years the Department has received significant enhancement funds for hiring new faculty in systems, graphics, HCI, and software engineering. It has received major new funding to support its undergraduate education program several times over the past 6–7 years. It (most importantly our previous chair, Prof. Gannon) has lobbied successfully for the construction of the new classroom facility for computer science education. A good amount of this funding has been lost during the current budget downturn, but the Department is in a far better position financially than it would have been without the additional support it received from the campus during the past five years.

5. **Graduate program not a high priority**. We have made significant changes (which we believe are improvements) to our graduate program. It is an area in which we need to make much more progress. The old written comprehensive exam system has been replaced by a course-based system. This, of course, solved some problems and introduced a whole new set. The program is under review once again based on discussions at our Department retreat in January 2001. But many of the deficiencies noted in the last review remain - mentoring of new graduate students (although we have instituted faculty/student lunches, seminars on academic careers, and other mechanisms to increase contact between new grad students and faculty), actively recruiting new graduate students, and, most important, placing our Ph. D.'s, are all areas in which significant improvement is still necessary.

6. **Joint appointments**. The last review suggested that additional joint appointments with ECE would be beneficial. This was symptomatic of a dearth of interdisciplinary research and educational activities involving computer science faculty. There has been astounding progress in this area over the past decade. Our faculty is now involved in an amazing variety of interdisciplinary research projects with faculty from Electrical and Computer Engineering, Nuclear Engineering, Linguistics, Sociology, Library and Information Sciences, Mathematics, Physics, Business and Management, and Geography. The table below lists our affiliate faculty, showing their home departments and the faculty in our department whom they work with. There are many additional relationships involving UMIACS faculty and others without appointments in either Computer Science or UMIACS.

Affiliate	**Home Department**	**Collaborates with**
John Baras	ECE	Roussopoulos
Rajee Barua	ECE	

Shuyra Bhatacharya	ECE	
Rama Chellappa	ECE	Davis, Aloimonos, Jacobs
Michael Cukier	Materials and Nuclear Engineering	Arbaugh, Basili, Zelkowitz
Allison Druin	Library and Information Studies	Bederson, Shneiderman, Gumbrietiere
Manoj Franklin	ECE	
John Grant	Towson State Math	Perlis
Satyandra Gupta	Mechanical Engineering	Nau
John Horty	Philosophy	Perlis
Joseph JaJa	UMIACS/ECE	Roussopoulos, Sussman
Gang Qu	ECE	
Louiqa Raschid	Business and Management	Roussopoulos, Dorr
Phil Resnick	Linguistics	Dorr
Uzi Vishkin	ECE	Tseng
Weinberg	Linguistics	Dorr
Donald Yueng	ECE	Tseng

Additionally, progress is being made on interdisciplinary undergraduate education programs. We recently completed an agreement with Physics for an undergraduate track in computational physics and are in the midst of the design of an undergraduate joint major in computer science and biology.

7. **Minorities**. The external review felt that the Department was not sufficiently active in recruiting minority students-both undergraduates and graduates. The University as a whole is now very successful at recruiting undergraduate minorities, and the Department still does not play any role in this process. Unfortunately, no steps have been taken to increase recruiting of minority graduate students.

In November 2001, 20 women and minority graduate students from the US and Canada came to our campus for a workshop entitled “Research, Careers, and Computer Science: A Maryland Symposium.” The participants presented their research, met Maryland faculty and students, and participated in interactive sessions about academic jobs. These sessions focused on teaching, research, obtaining funding, the tenure process, distinctions between academic and other jobs, and time management. This event was sponsored by both the Computer Science Department and UMIACS. More information can be found on the workshop website: http://www/cs.umd.edu/users/oleary/workshop/.

8. **Salary compression**. Our Department still suffers from severe salary compression, with some assistant professors having higher salaries than the lowest paid full professors. Starting salaries and startup packages are nationally competitive, but our senior faculty have fallen further behind since the last review. This is summarized in the table below that contains a Taulbee-like survey of Maryland salaries versus national statistics.

	Assistant Professor			Associate Professors			Full Professor		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
Maryland	79,500	84,402	93,600	67,915	94,194	105,300	92,427	120,524	172,869
National (1-12) means	74,711	80,891	86,483	84,148	91,412	97,949	88,632	122,732	168,860

(13-24)	78,070	83,673	90,538	85,663	92,985	98,827	94,322	127,845	185,306

Next, Faculty Research

Contents

Contents

Previous, Overview

Professorial Faculty and Research

The teaching and research interests of our faculty span most major areas of contemporary computer science. The Department has been consistently ranked highly in national surveys conducted by both the National Academy of Science and U.S. News and World Report (12'th in their most recent survey). Several specific research areas have been especially highly ranked (Software Engineering, Artificial Intelligence, Database Systems and Computer Vision), while other areas, we believe, have improved dramatically during the past ten years and should soon be nationally recognized (especially computer systems and computer security). Faculty research is supported by substantial external funding. This funding comes from federal agencies (NSF, DARPA, NIST, NASA, ONR, NIH, etc.), private foundations, and industry. At the end of 2002, our Department active awards totaled $16,364,240 in federal grants and contracts to the Department, $3,094,409 in gifts, and $51,642,160 in UMIACS awards (the latter are often collaborative efforts with faculty in other departments). Twenty proposals were submitted to the 2003 NSF ITR program; as that list shows, many of them included faculty form other departments and institutions.

Our faculty have been widely honored for their contributions to the field. We have

- 7 Fellows of the ACM,
- 4 Fellows of the AAAS,
- 5 Fellows of AAAI (American Association for Artificial Intelligence) and
- 11 Fellows of the IEEE.

Fourteen of our faculty have won NSF CAREER awards, four won its precursor, the PYI award, and one was named a Presidential Faculty Fellow.

Compared to ten years ago when our last review was conducted, our research program now has a significant interdisciplinary component, as discussed briefly in the overview section of this report. There are seventeen faculty from seven academic departments on our campus (Electrical and Computer Engineering, Linguistics, Philosophy, Mechanical Engineering, Library and Information Sciences, Business and Management, Materials and Nuclear Engineering) and one from the Mathematics Department at Towson State University who are actively engaged in joint projects with our faculty. Additionally, several of our faculty hold affiliate appointments in Electrical and Computer Engineering, and almost all of our faculty hold joint appointment with one of the two large computationally oriented research institutes on the campus – The Institute for Advanced Computer Studies and the Institute for Systems Research. There are, additionally, two recently established Laboratories in software engineering (the Fraunhofer Center) and computer systems (the MIND Laboratory) which afford our faculty opportunities to work closely with industrial partners on the application of Maryland technology to complex, real world problems. These, along with several other research institutes and laboratories with which our faculty have active collaborations, are described in an Appendix to this report.

The Department, in conjunction with UMIACS, has been very active in faculty recruiting during the past few years. In each of academic years 2000-2001 and 2001-2002 we interviewed over twenty candidates annually for positions in many areas and at all ranks. The tables below shows the faculty members we have hired since 1998. This year, because of budget limitations, we are searching for only two positions with UMIACS – a new UMIACS Director and a faculty member in the area of computational biology.

Assistant Professors

Faculty Member	PhD Received	Institution	Hiring Date
Bederson, Benjamin	1992	NYU	Jan-98
Chawathe, Sudarshan	1998	Stanford	May-99
Bhattacharjee, Samrat	1999	Georgia Tech	Aug-99
Arbaugh, William	1999	Penn	Jul-00
Memon, Atif	2001	Pittsburgh	Aug-02
Getoor, Lise	2001	Stanford	Dec-02
Iftode, Liviu	1998	Princeton	Jan-02
Sussman, Alan	1991	CMU	Jan-02
Hicks, Michael	2001	Penn	Jul-02
Guimbretiere, Francois	2002	Stanford	Aug-02
Katz, Jonathan	2002	Columbia	Aug-02
Foster, Jeffrey	2002	UC Berkeley	Jan-03

Associate Professors

Varshney, Amitabh	1994	North Carolina	Jan-00
Srinivasan, Aravind	1993	Cornell	Aug-01
Jacobs, David	1992	MIT	Aug-02

The Department places significant emphasis on providing a nurturing environment for its junior faculty. Startup packages, which cover summer salary, graduate student support, equipment, travel, etc., have averaged $250,000. Additionally, each new faculty member is assigned a mentor from the associate and full professors. This mentor assists the new faculty member with proposal preparation (reviewing drafts of proposals, especially CAREER award proposals), teaching strategies (dealing with academic dishonesty, rules for submissions and grading of projects, access to course materials from previous versions of the course), and as a general sounding board for questions and concerns. This program has been in place for the past two years. Teaching loads are competitive with other leading institutions. Our goal is to have faculty teach between 40-50% of their courses at the graduate level. New faculty are guaranteed 50% graduate courses during their first two years at Maryland. Special incentives are provided to recruit female candidates, including maternity leave benefits. Promotions to Full Professor have stringent requirements for national prominence in research and excellent in education and service. The table below summarizes promotions made since 1997.

Professorial Promotions (since 1997)

Faculty Member	Date	Title
Dorr, Bonnie	Aug-98	Associate Prof
Franklin, Michael	Aug-99	Associate Prof
Golubchick, Leana	Aug-01	Associate Prof
Hollingsworth, Jeff	Aug-00	Associate Prof

Keleher, Pete	July-01	Associate Prof
Khuller, Samir	Aug-97	Associate Prof
Porter, Adam	Aug-98	Associate Prof
Tseng, Chau-Wen	Aug-01	Associate Prof
Perlis, Don	Aug-97	Professor
Pugh, Bill	Aug-01	Professor
Saltz, Joel	Aug-97	Professor
V. S. Subrahmanian	Aug-00	Professor
James Hendler	Aug-00	Professor

The Department research environment is enhanced by a wide range of regularly ongoing conferences and seminars. These vary from Department Colloquia to many individual research group seminars, some of which are publicized (see Department Web page) and others which are not. A list of the Department Colloquia for 2002 is given below. Few if any Department Colloquia are held in the Spring as this is when faculty candidates come to give research talks, making for a full schedule of seminars.

Department Colloquia: 2002 Distinguished Lecture Series

Quantum Physics and the Nature of Computation
Umesh Vazirani, University of California-Berkeley

Peer-To-Peer Data Management
Hector Garcia-Molina, Stanford University

OceanStore: Toward Global Scale, Self-Repairing, Secure, and Persistent Storage
John Kubiatowicz, University of California-Berkeley

Immersive Virtual Reality for Scientific Visualization
Andries Van Dam, Brown University

Embedded Devices: The Impact on Optimizing Compilers
Mary Lou Soffa, University of Pittsburgh

Navigating Audio Like Text: Browsing and Searching Voicemail Messages
Julia Hirschberg, Columbia University and ATT Labs-Research

Challenges

Our Department of Computer Science is quite strong, and we believe it has the potential to move into the top ten in the US. Most members of our faculty have prominence in their research areas and several are acknowledged as among the world leaders in their fields. The size of the department coupled with the breadth of numerous faculty members adds to our strength. We see the following as the major challenges to meeting this goal

- We lost two of our database faculty members who specialize in database systems to top 5 universities (Faloutsos to Carnegie Mellon, Franklin to Berkeley). The recent hiring freeze will make it difficult to replace them, and to replenish that group.

- The University of Maryland separates Computer Engineering and Computer Science, with the former falling in the Engineering College (Department of Electrical and Computer Engineering). Thus, unlike at some of our peer departments, some areas associated with computing (particularly embedded systems and robotics) fall under the aegis of the engineering school. In recent years, a number of faculty have been given affiliate appointments between the two colleges, but we believe that a closer relationship would be desirable, and this will be a focus of activity in the coming years.

- The University has targeted computational biology as an area to expand, building on the talents of our faculty in both the computer science and the biology departments, and on our geographical proximity to the National Institutes of Health. The Computer Science Department is a bit concerned, however, that the University may be too narrowly focused on the computational aspects and would be better off taking a broader view of biocomputing, including bioinformatics and biorobotics and sensing.

- The recent construction of a new classroom building with facilities for networking, projection, and webcasting and containing significantly more space than our previous building, addressed a major infrastructure need for our department. Unfortunately, this increase in teaching space did not address the critical need for space for our research laboratories. In the past decade, the Department has significantly increased its presence in experimental computer science, but the lack of space makes it extremely difficult for groups to build the demonstration facilities necessary for many of our research endeavors. Recently, one of our laboratories has had to move off campus to provide housing for its students and research associates, and others threaten to follow. Space thus remains a critical need for our department.

- Although there is considerable demand for our graduate students, we believe our success in placing graduate students into academic positions is not commensurate with the general reputation of the department. We are actively exploring ways to improve the quality of students' research and presentations. This topic is addressed in more detail in the section on graduate education.

Next, Undergraduate Program

Contents

Interactions with other Departments and Institutes

UMIACS

The mission of UMIACS (University of Maryland Institute for Advanced Computer Studies) is to foster and enhance interdisciplinary research and education in computing across the College Park campus. Since its inception, UMIACS has played a major role at the University of Maryland in building strong interdisciplinary research programs, cutting-edge computing infrastructure, and long-term partnerships with national and international research centers. The Institute's programs are led by distinguished researchers, many of whom hold joint appointments in strong academic units such as Computer Science, Electrical and Computer Engineering, Linguistics, Geography, Philosophy, Business, Education, and College of Information Studies.

Since computing is at the core of all the Institute's activities, UMIACS has a uniquely close relationship with the Department of Computer Science. The synergistic environment provided by UMIACS enables innovative collaborations between the Computer Science faculty and other faculty on campus. These collaborations form the basis for several major research programs which are conducted through the various UMIACS laboratories, each lab being directed by a group of faculty and having its unique computing infrastructure. These laboratories provide a significant amount of support to our the Department in the form of funding for student salaries, equipment, and infrastructure for management of projects, employing undergraduates, and advising PhD. students toward the completion of their degree in the CS program. The key UMIACS laboratories are described below.

The Computational Linguistics and Information Processing (CLIP) Laboratory promotes research in machine translation, multi-lingual information retrieval, information mediation, and natural language translation. CLIP consists of two main entities: The Natural Language Group and the Database Group. The natural language group focuses on several areas of broadscale multilingual processing, e.g., machine translation, scalable translingual document detection, and cross-language information retrieval. The database group focuses on architectures for wide area computation with heterogeneous information servers, e.g., scientific discovery from biomolecular data sources. The Laboratory is co-directed by Bonnie J. Dorr, of the Computer Science Department, Louiqa Raschid of the Smith School of Business, and Amy S. Weinberg of the Linguistics Department. Other CLIP PI's are Philip Resnik (Linguistics) and Doug Oard (CLIS). Collectively, CLIP PI's have advised and graduated 4 CS PhD's, 4 CS Masters, and 9 CS honors students.

The Human-Computer Interaction Lab (HCIL) at the University of Maryland conducts research on advanced user interfaces and their development processes. Interdisciplinary research teams study the entire technology development life-cycle which includes the initial technology design, implementation issues, and evaluation of user performance. Currently, HCIL researchers are investigating new approaches to: information visualization, interfaces for digital libraries, multimedia resources for learning communities, zooming user interfaces (ZUIs), technology design methods with and for children, and instruments for evaluating user interface technologies. The Laboratory is directed by Ben Bederson of the Computer Science Department. Other HCIL PI's are Allison Druin (Education), Francois Guimbretiere (Computer Science), Jim Hendler (Computer Science), Kent Norman (Psychology), Doug Oard (CLIS), Gary Rubloff (Engineering), and Ben Shneiderman (Computer Science). HCIL PI's have collectively advised and graduated 7 CS PhDs, 5 CS Masters, and 2 CS Honors students.

The Language and Media (LAMP) Laboratory focuses on several areas of broadscale processing of natural language, e.g., document and video analysis, information access, machine translation, scalable translingual document detection, and cross-language information retrieval. Within these areas, the media analysis group

focuses on providing tools and techniques for access to large heterogeneous databases of multimedia information objects. Researchers working in document and video analysis have developed a number of prototype systems ranging from analysis of handwriting to compression to recognition of logos. Most recently, LAMP researchers have focused on automatic access to information sources by addressing issues involved in initial processing, organization, manipulation and retrieval. The Laboratory is co-directed by David Doermann (UMIACS) and Amy Weinberg (Linguistics). Other LAMP PI's are Daniel DeMenthon, Bonnie Dorr, Huiping Li, Doug Oard, and Philip Resnik. LAMP PI's have collectively advised and graduated 3 CS PhDs, 5 CS Masters, and at least 6 CS Honors students.

The Keck Laboratory for the Analysis of Visual Motion was established in 1997 through a grant from the Keck Foundation. The Laboratory serves as the basis for exploration of fundamental problems in the recovery of three dimensional models of human movements. The research conducted in the Keck Laboratory makes use of dynamic graphical representations of human movement and human manipulation of physical objects. It studies computer vision algorithms that analyze synchronized videos of activities obtained from multiple, calibrated viewpoints containing a combination of conventional and simulated biological sensors. These algorithms employ three dimensional geometric models of human body shape and of the geometry of the objects being manipulated, along with prior knowledge of the task to be performed and the physical objects involved in the task. Keck is directed by Larry Davis (Computer Science) and recently graduated 2 CS PhD's who investigated the use of three dimensional models to recover the time-varying articulation of a person in action.

The Distributed Systems Software Laboratory (DSSL) provides a collection of UNIX systems for conducting systems research. It consists of a collection of personal computers, 2-4 multi-processors (using AMD, PowerIII, and MIPS processors), an easily re-configurable collection of Ethernet switches, and high speed networks (such as Myrinet). Projects using lab resources include Active Harmony (a system for building runtime adaptable programs), Dyninst (a runtime binary editing tools), MoteFS (a remote filesystem running over untrusted wide-area networks), NICE (a cooperative framework for scalably implementing distributed applications over the Internet), and TerraDIR (a distributed peer-to-peer directory protocol).

The Graphics and Visual Informatics Laboratory (GVIL) was established in 2000 to promote research and education in computer graphics, scientific visualization, and virtual environments. The mission of GVIL is to improve the efficiency and usability of visual computing applications in science, engineering, and medicine. The scope of this laboratory's research covers design of algorithms and data structures for reconciling realism and interactivity for very large graphics datasets, rapid access to distributed graphics datasets across memory and network hierarchies, and study of the influence of heterogeneous display and rendering devices over the visual computing pipeline. The activities of the laboratory involve development of visual computing tools and technologies to support the following research-driving applications: protein folding and rational drug design, navigation and interaction with mechanical CAD datasets, and ubiquitous access to distributed three-dimensional graphics datasets. GVIL is directed by Amitabh Varshney (Computer Science) who is currently advising 6 CS PhD students. GVIL has graduated 3 honors CS students.

The Perceptual Interfaces and Reality Laboratory (PIRL) is investigating perceptual Interfaces for extending human computer interaction to use all modalities of human perception. Current research efforts are focused on including vision, audition, and touch in the process. The goal of perceptual reality is to create virtual and augmented versions of the world, that are perceptually identical to the human with the real world. The goal of creating perceptual user interfaces is to allow humans to have natural means of interacting with computers, appliances and devices using voice, sounds, gestures, and touch. PIRL researchers are also focused on the

creation of prosthetic devices for the vision and hearing impaired, by mapping inputs from one modality into equivalent ones in another, so that computationally augmented input streams can be created with extra content from the missing modality. PIRL is directed by Ramani Duraiswami (UMIACS). Other PIRl PI's are Larry Davis (Computer Science), Nail Gumerov (UMIACS), Yaser Yacoob (UMIACS), and David Harwood (UMIACS). PIRL PI's have collectively advised and graduated 2 CS PhDs, 1 CS Masters, and 1 CS Honors student.

Fraunhofer Center for Experimental Software Engineering, Maryland

In Germany, the Fraunhofer Gesellschaft is a society of 55 applied research institutes with a staff of 11,000 employees. Each institute works with government and corporate clients and specializes in a specific technology including electronics, materials, lasers, and computing. Seven of the institutes have a presence in the United States as divisions of the non-profit company Fraunhofer USA. The Fraunhofer Center, Maryland was started by Computer Science professors Victor Basili and Marvin Zelkowitz and is affiliated with the Fraunhofer Institute for Experimental Software Engineering in Kaiserslautern, Germany. The Kaiserlautern Institute director Dr. Dieter Rombach is a former professor in the Computer Science Department of the University of
Maryland. The Center occupies about 5,000 square feet of space just south of the campus in College Park. The Fraunhofer Center currently has 12 full-time and 5 part-time employees (including several university faculty) and several students and visitors work there. Dr. Basili is the Executive Director, Dr. Zelkowitz the Chief Scientist, and Dr. Frank Herman the managing Director.

Fraunhofer USA and the University of Maryland work cooperatively on software engineering research issues. The center is partially supported by base funding (approximately 27% of 2002 funding comes from the state of Maryland and the Fraunhofer Gesellschaft in Germany). The remainder comes from various contracts and grants that the Center must secure. As both a research and business strategy, the Center and the Experimental Software Engineering Group of the Department work cooperatively on many projects. Weekly group meetings are held with University graduate students and faculty and Fraunhofer scientists jointly discussing research activities. Many joint projects have been funded, with the Fraunhofer Center usually a subcontractor to the University. The Center was a participant in the Department's NASA Software Engineering Laboratory from 1999 through 2001, as well as being a current participant in both the Center for Empirically Based Software Engineering (CeBASE) NSF ITR and the NASA Ames Research Center funded High Dependability Computing Program (HDCP) grants. The HDCP grant is a 5-year program, with the Department and the Center subcontractors to Carnegie Mellon University.

The Center complements and extends the core competencies of the University's software engineering group. Measurement, experimentation, experience factory software development, COTS development, and reading and inspection technologies are all core competencies of both the University and Center. In addition the Center is developing expertise in knowledge management and agile development. The Center works with a variety of international corporations, including Boeing, Motorola, Daimler Chrysler, Nokia, and ABB, as well as many smaller companies in Maryland.

The strategy of the Center and Department working together provides opportunities, especially in software engineering, that would be difficult to accomplish solely in a university. Software engineering needs large laboratories, which in this context are software development projects within the industrial or government sector. The Center provides the means to provide on-site staff and the means to work on projects over many years that is difficult to do with university students, and the Department provides a basic research focus that is difficult to maintain within a corporate setting. For example, one Center project for the US Army is looking

at deployment in 2008 and the HDCP grant is looking at the next NASA Mars mission in 2010. These would both be difficult to work on solely in a University context.

MIND Laboratory

The Maryland Information and Network Dynamics (MIND) Laboratory was established to provide a focal point for defining, developing, evaluating, and deploying new information technologies through innovative methods. By collaborating with industry and federal agencies, the MIND Lab works to invent new ways of meeting technological challenges by addressing key research issues in the fields of education and information technology. Fujitsu Laboratories of America is a Founding Partner of the MIND Lab and has established their east coast laboratory in College Park. The Aerospace Corporation, Lockheed Martin Advanced Technologies Laboratory, and Koolspan, LLC have joined as Affiliate Partners.

The MIND Lab brings together a world-class group of researchers from the University of Maryland's Department of Computer Science to work with industrial leaders who are facing the challenges of explosive user-demand for immediate access to broader, more sophisticated and integrated information services. Drs. Ashok Agrawala, A. Udaya Shankar, and Bobby Bhattacharjee work on technological issues related to networking and pervasive computing. Dr. James Hendler is the MIND Lab's semantic web expert, and Dr. William Arbaugh focuses on security. Other faculty members in the Department also share their expertise as new projects are defined. The MIND Lab employs several graduate students in the Department of Computer Science. This experience gives the students critical real-world experience with some of the world's most dynamic companies. The MIND Lab has graduated 4 PhDs, 2 MS students, and 2 BS Honors students from the Department of Computer Science and continues to fund 7 graduate students who are working toward their degrees.

The MIND Lab is a joint effort between the Department of Computer Sciences and the University of Maryland Institute for Advanced Computer Studies (UMIACS). The MIND Lab initially received start-up funds from both for this venture. Dr. Ashok Agrawala, Director of the MIND Lab, has been a faculty member in the Department of Computer Science for thirty-one years.

Applied Mathematics and Scientific Computation Program

The Program in Applied Mathematics and Scientific Computation (AMSC) is an interdisciplinary program that offers Ph.D. and M.S. degrees as well as a post-baccalaureate Certificate in Scientific Computation. The program is a reorganized version of what had previously been the
Applied Mathematics program, reflecting an increased emphasis within the program on computational techniques in science and mathematics. Graduate students can pursue degrees in either of two tracks: a concentration in Applied Mathematics, which combines a foundation in mathematics together with advanced study and research in an area of application; or a concentration in Scientific Computation, in which students are trained in the use of computational techniques and associated information technology with correspondingly less emphasis on formal mathematical methods.

The program has 120 faculty members from nineteen departments and Institutes. Thirty-three members of the CMSC faculty are affiliated with the program. The members of the Numerical Analysis and Scientific Computing Field Committee (Elman, O'Leary and Stewart) are actively involved in most aspects, including direction of PhD students (six completed their degree in the past five years) and MS students; helping in the design of the Scientific Computing curriculum; teaching courses, including new graduate courses in Scientific Computing (AMSC/CMSC 660-661) and Computer Organization and Scientific Computing (AMSC/CMSC 662); directing of projects for AMSC 663-664, a new required independent study course;

writing and grading of the Numerical Analysis qualifying exam; serving on the Graduate Committee on Applied Mathematics, which s responsible for evaluating student thesis prospectuses and performance on comprehensive exams; and erving on committees for related programs, such as the search committee for the CSCAMM director.

Other computer science faculty play a lesser but still significant role in AMSC, most notably through advising of students. For example, Jim Reggia is typically advising one AMSC doctoral student, Larry Davis (in collaboration with Ramani Duraiswami from UMIACS) currently has several advisees, and Bonnie Dorr has had two advisees in the last five years (one current).
Several faculty also occasionally serve on students' advisory committees.

Institute for Systems Research (ISR)

ISR was formed in 1985 as a joint venture between the University of Maryland and Harvard University, as an interdisciplinary National Science Foundation (NSF) Engineering Research Center with the specific mission of conducting systems research. Since 1985, ISR's vision has evolved from one concerned with computer-aided tools for design of automation and information processing systems to its current broader concern with the integrated design for control of complex engineering systems that have control and communication systems as subsystems.
ISR is an acknowledged leader in the integrated design for control of complex engineering systems. It has linked 11 departments at the University of Maryland and Harvard University.
In eight years, over 100 companies have benefited from participation in ISR programs, and 16 internationally renowned research organizations have established formal exchange agreements with ISR. In 1992, the state of Maryland recognized the unique contribution of ISR by designating it a permanent institute of the University of Maryland and renewing its financial commitment to the Institute. ISR occupies 27,000 square feet of space in facilities at the University, which provides space for eight constituent and five affiliated laboratories. A sixth affiliated laboratory is located at Harvard University.

One of the ways in which ISR fosters interdisciplinary research is that all ISR faculty appointments are joint with various academic departments. The following CS faculty have affiliations with ISR, either as joint appointments or affiliate appointments: James A. Hendler, Nicholas Roussopoulos, Ben Shneiderman, V.S. Subrahmanian, and Dana S. Nau. Through ISR, CS faculty have worked on joint research and educational projects with faculty from Mathematics, Business, and nearly all of the departments in the School of Engineering. Several of these projects have been quite successful in terms of their visibility and impact.

In addition to its cross-disciplinary research programs, ISR has a cross-disciplinary educational program. Several of our faculty have supervised MS students in ISR's systems engineering graduate program. ISR administers the Gemstone program, a cross-disciplinary undergraduate honors program, and several CS faculty have supervised groups of Gemstone students in multi-year cross-disciplinary educational projects (see Undergraduate Education section above).

Neuroscience and Cognitive Science Program

The Neuroscience and Cognitive Science (NACS) Program at UMCP is a multi-disciplinary program that offers research and training opportunities in neuroscience, cognitive neuroscience and computational neuroscience. The program involves not only faculty on the UMCP campus, but also on two other University of Maryland campuses (Baltimore City and Baltimore County). Not only does the NACS program offer graduate training leading to a Ph.D. degree, but it also helps to coordinate cross-disciplinary research (e.g., via its colloquium series). Students getting a PhD in the NACS program have access to a broad range of core

graduate courses (e.g., Introduction to Neuroscience, Cognitive Neuroscience, Computational Neuroscience). Each student selects a ``home department'' and may take courses in that department to complete program requirements. For example, some NACS students have taken CMSC 727: Neural Modeling for this purpose. At present several faculty in Computer Science are active members of this program (Dorr, Hendler, Perlis, Reggia and Smith), and occasionally serve as the primary PhD advisor for the computationally-oriented students. Further information about this program can be found at the NACS web page: http://www.inform.umd.edu/nacs/

Contents

Faculty Profile

This appendix summarizes the research interests of our faculty. While the research program is highly interdisciplinary, we have organized this section according to the major core computer science research areas in which we have critical mass. These include:

Algorithms and Theory of Computation
Artificial Intelligence
Databases
Computer Vision and Graphics
Scientific Computing
Programming Languages and Software Engineering
Human-Computer Interactions
Systems and Networks

As some faculty conduct research in multiple committees, and as there is widespread collaboration within and between areas, some faculty's research is split into multiple parts below for clarity.

Algorithms and Theory of Computing

Faculty: Gasarch, JaJa, Katz, Kruskal, Khuller, Mount, Smith, Srinivasan and Vishkin

This group's research interests are broadly in the areas of algorithms, logic, complexity theory, cryptography, computational geometry, data structures, learning theory and parallel computing. Several faculty in this group have been working with other faculty in application areas such as graphics, security, networking, image processing and information retrieval. Over the last five years, we have hired two new faculty (Katz and Srinivasan), and published numerous journal and conference papers, including ten papers in the most prestigious and highly competitive FOCS and STOC conferences. In addition, many papers have been published in other highly selective conferences such as SODA, ICALP, CRYPTO and Computational Geometry. Many of our theory faculty have given invited talks at conferences and other Universities about our research.
For the first time, Maryland has two faculty members in the ACM STOC 2003committee. In addition, faculty have *chaired* the committees for several other conferences recently (ACM CG Conference,Workshop on Algorithms Engineering and Experiments, as well as the Workshop on Approximation Algorithms).

Bill Gasarch studies primarily complexity theory and games. Complexity theory's ultimate goal is, given a problem, determine how hard it is under a variety of measures. This involves logic and combinatorics. Games are of interest for math and computer science education as they can be used to illustrate many concepts in a fun way.

Joseph JaJa's research focuses on development of efficient algorithms and indexing schemes for large scale spatial OLAPs, and for designing efficient scalable algorithms that are portable across different

multiprocessor architectures, and on development of efficient data structures and algorithms for range search queries on spatio-temporal data.

Jonathan Katz studies cryptography, network/computer security, and distributed algorithms. One recent representative project is the design of efficient cryptographic algorithms which remain secure in the face of key exposure. He has explored cryptosystems in which the secret key evolves over time, and for which exposure(s) of the secret key corresponding to some time periods have minimal effect of the security of the system at all other times. As part of this work, he has constructed the first forward-secure public-key encryption scheme (answering a problem which had been open for 5 years), introduced the notion of "key-insulated security" and designed encryption/signature schemes secure in this sense, and designed the first intrusion-resilient public-key encryption scheme. Other work in this area has included the development of new threshold cryptosystems and consideration of "forward-secrecy" in password-authenticated key exchange.

Samir Khuller studies efficient algorithms for solving NP-hard problems. These algorithms produce sub-optimal solutions. The goal is to show that the cost of the solution produced is guaranteed to be close to optimal. Such analysis often leads to a better understanding of the problem as well as the development of general algorithmic methods for optimization. The specific problems have to do with issues of load balancing in storage servers, as well as data movement and re-organization in Storage Area Networks, which are very large storage servers that are widely used for serving multimedia data. Algorithms have been developed and analysed both from the worst case point of view, as well as the average behavior as observed experimentally. Several methods have been developed and patent applications are currently pending. In addition, Dr. Khuller has been exploring data movement applications over the Internet as well, via the use of network flow based methods in the context of the Bistro Project (a system to upload data over wide area networks such as the Internet).

Clyde Kruskal studies parallel computers and parallel algorithms. In addition, he has been working on discrete geometry problems related to coloring the plane.

David Mount investigates the design of efficient algorithms and data structures for geometric problems. His principal interest has been problems with applications in areas such as image processing, pattern recognition, information retrieval, and computer graphics. The major thrust of his recent work has been in approximation algorithms. This work is motivated by applications in which existing exact algorithms are unacceptably slow or use excessive space. He has shown that by accepting a small error relative to exact solutions, it is possible to achieve significantly faster algorithms using relatively simple algorithms and data structures. Examples of problems for which such efficient solutions have been found include nearest-neighbor searching in multidimensional Euclidean spaces, clustering multidimensional data sets, performing pattern matching and registration in digital images, and robustly fitting lines and curves to noisy data. These projects have resulted in theoretical advancements, and they have also been publicly released as software systems, such as ANN, a C++ library for approximate nearest neighbor searching.

Carl Smith studies algorithmic learning theory, the process whereby computers learn from examples; discovery science, the process by which computers can automatically discover interesting facts from huge data sets; and models of computation, a study of computation from a variety of view points.

Aravind Srinivasan conducts research on the design and (theoretical/experimental) analysis of algorithms with applications in networking, combinatorial optimization, information retrieval, and related areas, with an

emphasis on probabilistic methods. He has developed rigorous approaches for the design and analysis of approximation algorithms through randomization; the focus here has been on efficiently solving a ``relaxation'' of a given (difficult) combinatorial optimization problem, and using randomization to restore the violated constraints. This has led to the current-best approximation algorithms for low-congestion packet routing and scheduling, scheduling broadcasts in pull-based systems, minimal-redundancy storage in distributed networks, etc. He has also developed algorithms for security and multicast in peer-to-peer networks. His interests also include distributed algorithms: a representative recent result is on fast distributed algorithms for constructing ``backbones'' for communication in wireless ad hoc networks.

Uzi Vishkin studies parallel algorithms and architectures. The ongoing PRAM-On-Chip project is a direct outgrowth of theoretical work. The project offers a concrete agenda for challenging the 1946 von-Neumann architecture through streamlining the massive knowledge base developed by the parallel algorithms community with the roadmap for CMOS VLSI.

Artificial Intelligence

Faculty: Dorr, Getoor, Hendler, Nau, Perlis, Reggia, Subrahmanian

The AI research group at the University of Maryland has major strengths in several diverse areas. These include multilingual processing. AI planning, software agents, ontologies, logic in AI, machine learning, biologically-inspired systems, and uncertainty management.

Bonnie Dorr and her colleagues focus on several areas of broadscale multilingual processing, e.g., machine translation, scalable translingual document detection, and cross-language information retrieval. They have investigated the problem of creating new statistical models that are linguistically informed, leading to higher quality output for a wide range of languages while still being practical to train and use. She has developed the technique ``divergence unraveling'', i.e., the detection and resolution of language-pair mismatches where the realization of a concept is distributed differently in different languages. This technique is currently being explored as a vehicle for translating into English from languages as diverse as Arabic, Chinese, and Spanish. This approach (and sub-components thereof) is being tested in a wide range of applications including statistical machine translation, interactive cross-lingual browsing, and headline generation.

Dana Nau, Jim Hendler and V. Subrahmanian are working in AI planning. The motivation for this research has been to develop synergy between the theory and practice of planning: better theories of planning can lead to more useful planning algorithms for practical applications, and experience developing planning systems for practical applications can lead to better theories of planning. This work has produced award-winning results in many different areas. For example, development of planning techniques for the Bridge Baron computer program enabled it to win the 1997 world championship of computer bridge, our SHOP2 planning system won one of the top four awards at the AIPS-2002 International Planning Competition, and our paper on the complexity of plan adaptation using derivational analogy won the award for the best research paper at the ECCBR-2002 conference.

V. Subrahmanian and Jim Hendler are conducting research on software agents. The primary goal is to develop an infrastructure for the creation and management of large scale agent applications. The IMPACT project has developed such a platform where agents can be built on top of legacy code. This work has also developed a firm theoretical foundation for such agents, and extended this theory to handle agents that reason about time, uncertainty, about the beliefs and possible actions of other agents, and security and

survivability properties of agents.
This work is closely tied to the ontology and uncertainty work listed below, and the planning work listed above. IMPACT has been used in several DoD applications.

Jim Hendler is one of the creators of the Semantic Web vision and is leading an active research group working on ontologies. This vision focuses on making more of the content on the web machine-readable, and indexing that content against ontologies that enable humans and machines to more precisely define shared terminologies. Beyond ontology, the Semantic Web focuses on logics, proofs, and the development of models whereby Semantic Web agents will be able to determine the trustworthiness of information found on the web and/or communicated from other agents.

Don Perlis works in the areas of common sense reasoning, cognitive modeling, and computational theories of the conscious mind. The underlying methods that he brings to bear on all of these are principally those of metareasoning and time-sensitive reasoning. Together these two provide very powerful tools for tackling many real-world aspects of intelligence, including reasoning in the presence of contradictory information, and with applications to natural-language human-computer dialog, planning with tight deadlines, and recognition and repair of mistakes.

Lise Getoor is investigating the use of structured statistical models for discovering link patterns in graphs, extracting information from semistructured sources and schema discovery and reformulation. She has done work in learning statistical models from both relational and semistructured data. She has successfully shown their effectiveness for selectivity estimation for database query optimization. She has also shown how simple models of link distributions in graphs can be used to improve classification accuracy. She is currently investigating richer models of link distributions, with applications to mining graphs such as the world wide web, citation graphs and social networks. She is working with researchers at ISI/USC on the problem of wrapper induction using structured statistical models. She is also currently involved in a NIMA project on visualizing changing patterns in large graphs.

Jim Reggia and his students are working on three aspects of biologically-inspired computation and AI. First, in computational neuroscience, his research group is studying neural models with a goal of understanding their self-organization, response to sudden damage, the processing of temporal sequences, and neural network learning in general. Second, in evolutionary computation, this group has been using genetic algorithms and related methods to evolve neural networks (as a means of better understanding biological neural circuits and architectures) and multi-agent communication systems in artificial worlds. Finally, cellular automata models of self-replicating machines have been developed to study the fundamental information processing principles involved in self-replication, and how self-replication might arise spontaneously from non-replicating components.

V. Subrahmanian is interested in uncertainty management. He has developed logic programming languages to handle uncertainty, as well as uncertainty and time. In addition, he has developed extensions of the relational algebra to handle uncertain reasoning in relational DBMSs and object oriented DBMSs. Database support for temporal uncertainty has also been studied. More recently, Lise Getoor and V. Subrahmanian developed XML models that incorporate uncertainty, and Dana Nau and V. Subrahmanian developed methods both to plan in the presence of uncertainty as well as to provide database support for monitoring plans, and creating new plans in the presence of existing corpora of plans.

Databases

Faculty: Chawathe, Getoor, Raschid, Roussopoulos, Samet, Subrahmanian

Sudarshan Chawathe's primary research interest is semistructured data, which is data whose structure is irregular, incomplete, and frequently changing. Examples of such data include structured documents (e.g., memos, legal briefs, forms) and data obtained by integrating disparate information sources (e.g., Web sites). With the adoption of XML, semistructured data is growing in both quantity and variety. Semistructured data (and its XML serialization) is modeled as an edge- and node-labeled rooted graph. His SEuS data mining system for efficiently determining frequent substructures in graph data is based on using a concise summary of data as a filter to prune the search for frequent structures. This approach is several orders of magnitude faster than comparable methods, and can efficiently process gigabytes of disk-resident data. He is also working on methods for processing streaming data, which is data that is accessible only in a serial order determined by the data source. Each data item is presented only once. It is not possible to seek forward or backward in the data stream, and data cannot be recalled unless explicitly buffered. He has developed automaton-based methods for evaluating XPath queries on streaming XML data, and implemented them in the XSQ system. He is currently working on expanding this work to more powerful query languages for streaming data. He is also working on differencing, summarization, and visualization of graph-structured data.

Louiqa Raschid's interests include architectures for wide-area computation with heterogeneous information servers; publishing and locating sources based on quality and content metadata using the Web and XML; and query planning and optimization. Wide-area applications utilize the wide-area network to connect hundreds of servers with thousands of clients. Such applications face significant challenges. This dynamic network may result in a wide variability in end-to-end latency. Similarly, as cached resources become obsolete, the staleness of delivered information may vary. Her research is undertaking a comprehensive study of the changing behavior of resources. Her objective is to develop appropriate resource profiles to characterize this behavior and to use these profiles to customize service and information delivery to clients. She is also working on managing the rapidly growing and diverse datasets available to the biological enterprise. Such data presents significant opportunities and challenges for data integration and seamless access. Her research is applying prior expertise with developing data integration architectures based on wrappers and mediators to provide seamless access to heterogeneous Web accessible sources. She is developing techniques from areas such as query optimization, adaptive query evaluation, machine learning and schema mapping and integration of heterogeneous databases, to solve problems of data integration.

Nick Roussopoulos' recent research focuses on data warehousing, dynamic Web content, and network intrusion detection. This work has developed storage architectures and indexing for efficient computation and management of data cubes generated from very large multidimensional data sets. Data cubes provide summaries and aggregations of all possible views of data, and their sizes grow exponentially. The latest result of this storage technology, the Dwarf Cube, obtains a data-reduction reduction up to one million to one. This technology is being evaluated by the largest database companies. The WebViews project aims at improving the performance of database-backed Web servers which are commonly used to generate dynamic content on the Web today. Such content drains Web server resources and inhibits scalability. This work has shown that by using our WebView technology, servers can be scaled up to two orders of magnitude without sacrificing the timeliness of the served information. Nick Roussopoulos is also studying data acquisition for network intrusion detection. The objective of his effort is to design an efficient, adaptive, and decentralized vent data acquisition system for tactical intrusion detection.
Techniques are being developed in three basic areas: (a) online correlation and value dependency discovery

over stream data using a limited amount of memory and instructions per data item, (b) compression of data cubes for log data, and (c) delivery of compressed aggregated/correlated data under limited bandwidth.

Hanan Samet's primary research focuses on spatial databases. A significant part of this research effort deals with the integration of spatial and nonspatial data into a database management system. The SAND Spatial Browser is an example of such work which enables users to pose queries that combine spatial and nonspatial data, with the spatial components of the queries being specified graphically. A Java-based version of the SAND Spatial Browser that is designed to enable its use over the internet has been made available to the outside world, and is also being investigated in a peer-to-peer environment. The ultimate goal of the SAND Spatial Browser project is the creation of a spatial spreadsheet. One of the most prominent features of the SAND Browser is the ability to perform ranking. This is an operation that enables us to retrieve data in the order of their distance (spatial) from another spatial object and we can also restrict the domain from which the data is drawn. The approach used differs from conventional approaches in the sense that it is incremental. In essence, once we have obtained the k nearest items, if we want the next nearest, then we can obtain it directly. This approach works for arbitrary geometric objects rather than being restricted to points, and also works in a metric space. It has been shown that this algorithm is I/O optimal. This algorithm has also been used in the VASCO System of Java applets which enables the visualization of over 30 different spatial data structures and algorithms for basic spatial operations that make use of them.

V.S. Subrahmanian is working on several major topics in databases as well as AI. First, he is developing ontology-based models for semantic integration of diverse, distributed heterogeneous databases. Second, he is developing models of databases (relational, object oriented and semistructured) to represent and manipulate time, uncertainty and spatial data. Third, he is developing models of databases to support sophisticated AI planning applications. Fourth, he is developing extensions of the relational algebra to query and summarize multimedia data sets. Finally, he is focusing on efficiently scaling heterogeneous databases via a variety of techniques.

Computer Vision, Computational Geometry and Graphics

Faculty: Aloimonos, Davis, Jacobs, Varshney, Samet

Faculty in this area perform research in computer vision, computer graphics and data structures for geometric objects. The Department has traditionally been very strong in the area of Computer Vision. The Computer Vision Laboratory is one of the oldest and largest laboratories of its kind in the world, performing research in almost all computational aspects of visual processing, both in theory and applications. In addition, the group draws strength from its association with other faculty, such as David Mount (computer Science) who works on Computational Geometry and Rama Chellappa (Electrical Engineering) who works in Signal Processing and many aspects of Computer Vision.

Yiannis Aloimonos' research during the last five years has concentrated on the problem of visual motion, and specifically on the recovery of three-dimensional models of the world from multiple views. One basic result obtained is the relationship of the error in building 3D models from video to the field of view of the camera. As a result sensors with a full field of view can better estimate 3D motion and thus acquire better 3D models. This has led to several efforts in building a variety of panoramic sensors. In Aloimonos' lab, the geometry of eye design has led to an understanding of and developed two principles governing eye design as it relates to the ability to recover 3D models from the processing of the images acquired by the eye. An outcome of this study was the understanding of geometric constraints characterizing the moving plenoptic function. Finally, a

new mathematical framework was introduced recently under the heading of Harmonic Computational Geometry as a tool for addressing the correspondence problem (matching) by relating properties of the signal to the 3D geometry. This work bridges the gap between harmonic analysis (signal processing) and geometry.

Larry Davis's principal research area is computer vision, with a focus on visual surveillance. He and his colleagues have been developing new vision algorithms and systems for detection and tracking of people from collections of fixed surveillance cameras, and for analysis of their actions and interactions. His most recent work in surveillance involved the development of kernel density estimation techniques for detection using background models and tracking using combined spatial-color models; multiperspective Bayesian methods for 3D detection and tracking of people in cluttered environments; methods for fitting 3D density models of articulated objects to 3D volumetric reconstructions; and various methodologies for detection of people from moving camera platforms.

David Jacobs has primarily worked on object recognition, given that he sees perceptual organization, object identification, and visual memory as all part of this process. His work on perceptual organization has focused on developing Markov models of shape to assist perceptual completion, and studying human perceptual organization. Work on object identification has focused on understanding how changing lighting effects the appearance of objects, and on how changing pose affects their apparent shape. Work on visual memory has attempted to model human memory, and build retrieval systems for 3D shapes. Some key results on these problems over the last five years are: analytic explanation for the fact that the images of a Lambertian object lie in a low-dimensional linear subspace, and the experimental use of this fact in a face recognition system; development of a photometric stereo algorithm for images with complex, unknown lighting; methods of comparing images that judge how similar they are in a way that is insensitive to lighting changes; a novel model of human word memory, with experiments on human subjects to judge its applicability; methods for finding the pose of objects using correspondences between regions, rather than local geometric features; an affine structure-from-motion system that handles points that appear or disappear from view in the middle of the motion sequence; and development of a search engine that uses shape similarity to find 3D computer graphics models.

A. Varshney's research in the Graphics and Visual Informatics Laboratory deals with a range of issues in visual computing. These include multiresolution modeling and rendering including view-dependent and view-independent simplification hierarchies as well as topology-preserving and topology-reducing hierarchies. The goal here is graphics acceleration with minimal visual degradation as well as enabling 3D graphics over low-bandwidth networks. Recent work also includes developing a lighting model to incorporate subsurface scattering effects within the local illumination framework. In the area of computational biology, he is developing visual informatics tools and technologies that will give scientists deeper insights in understanding the relationships between form and function in various biological proteins. He is developing new methods for efficiently computing and displaying electrostatic potentials by explicitly generating and incorporating the solvent interface. Among many factors involved in protein-protein interactions, shape complementarity is of major importance. The goal of ongoing research in shape complementarity is to develop fast and reliable methods for finding docking sites and corresponding transformations to align the two molecules into complementary conformations. In research on display technologies, he has built a tiled-display system that achieves geometric alignment for 3D graphics applications by pre-warping 3D objects. An ultrasonic tracker is used for user interactions with the displayed objects.

Hanan Samet has developed a map recognition system that combines image, locational, and nonspatial data

in a database. The image data in his case consists of a tourist map of Finland and the goal is to locate the various tourist sites by use of their representative symbols on the map. This works involved preprocessing the data and inserting it into the database so that it can be retrieved. Two approaches are used. The first classifies all of the symbols a priori and inserts a classification into the database. The second, stores a feature vector for each symbol. Thus the distinction is somewhat like static versus dynamic binding. Hanan Samet has also developed a method of graphically specifying the queries that uses pictorial query trees. The leaves of a pictorial query tree correspond to individual pictorial queries that specify which objects should appear in the target image, spatial constraints on the distance between them and their relative position, and the minimum required certainty of matching between query-image objects and database-image objects. This work has been generalized to enable users to specify the query shapes instead of being restricted to predefined symbols such as those found in a map's legend. In particular, the ability to handle rectangle, polygon, ellipse, and B-spline shapes was added. The retrieval process makes use of an abstraction of the contour of the shape which is invariant against translation, scale, rotation, and starting point that is based on the use of Fourier descriptors. These abstractions are used in a system to locate logos in an image database.

Scientific Computing

Faculty: Elman, O'Leary, Stewart

Numerical analysis at Maryland has been an extraordinarily strong area since the 1960s and remains quite strong. Werner Rheinboldt, an internationally known expert in the numerical solution of nonlinear equations, was director of the Computer Center and a founding member of the Computer Science Department. Our focus involves algorithms and applications in the fields of numerical linear algebra, partial differential equations, and optimization. Our faculty are active participants in the Applied Mathematics Program at Maryland, which was ranked 11th in the 2002 US News & World Report review of graduate programs. The Mathematics Department at Maryland also supports numerical analysis, with a focus primarily on numerical solution of partial differential equations. Faculty there includes John Osborn, C. David Levermore, Ricardo Nochetto, Tobias von Petersdorff, Bo Li, and Jian Guo Liu. Our relations are close, with most important decisions being made by a joint field committee. We run a weekly seminar in cooperation with Mathematics, IPST, and UMIACS, and we compensate for our small size by hosting visitors and postdocs. We have research funding from ONR and NSF. During the last 5 years we have received a best paper award, an outstanding poster award, and Stewart was the recipient of the Bauer prize from the Technical University of Munich. We are members of editorial boards of the top journals in the area

- Linear Algebra and Its Applications,
- SIAM Journal on Matrix Analysis and Applications,
- Siam Review,
- Numerische Mathematik,
- Mathematics of Computation
- SIAM Journal on Scientific Computing

Pete Stewart has published two volumes of a projected 5-volume series on *Matrix Algorithms: Basic Decompositions* (1998) and *Eigensystems*, (2001). We have published 38 journal articles during 1998-2002, with 6 more scheduled for publication.

Howard Elman, Diane O'Leary and Pete Stewart are all investigating various aspects of

Krylov subspace methods for solving linear equations and eigenvalue problems. Krylov subspace methods (e.g., the conjugate gradient and GMRES methods for solving linear systems and the Lanczos and Arnoldi methods for the eigenproblem) are the workhorse algorithms for large matrices, and our group has contributed to the understanding of these methods. Elman and O'Leary, have studied problems for which GMRES makes no progress in its initial iterations. The tool for analysis was a nonlinear system of equations, the stagnation system, that characterizes this behavior, and we developed several new results on when and why matrices stagnate. O'Leary and Stewart are pursuing some new phenomena in the convergence of Krylov subspaces with error. These new techniques promise to drastically reduce the work involved in some variants of these methods

Elman, O'Leary, and colleagues have developed a multigrid algorithm for numerical solution of acoustic scattering problems that overcomes the breakdown of the standard algorithm as the wavenumber in the Helmholtz equation increases. They also developed efficient numerical methods for solving the Helmholtz equation when the data or the forcing function is stochastic.

O'Leary, Stewart, and their students have studied algorithms for computing regularized solutions to ill-posed problems, with emphasis on the restoration of blurred images. Recent work has focused on blind deconvolution (where the blurring function is not known exactly) and on tools for computing and displaying the uncertainty in reconstructed images (joint work with James Nagy at Emory).

Howard Elman and his collaborators have developed new algorithms for solving the incompressible Navier-Stokes equations. The approach is based on preconditioning methodologies that take advantage of the saddle point structure of the problem and account for couplings of physical quantities (velocities and pressure) while maintaining computational efficiency.

Diane O'Leary and her collaborators have worked on document retrieval and summarization through linear algebra. For retrieval, she developed the semi-discrete matrix decomposition for use in latent semantic indexing (LSI). For summarization, she uses hidden Markov models plus a pivoted QR decomposition.

Pete Stewart and a student are developing a Fortran95 wrapper called Matwrap for matrix operations that should make it easy to turn code from matrix oriented languages, such as MATLAB, into highly efficient programs.

Programming Languages, Software Engineering

Faculty: Basili, Bederson, Foster, Guimbretiere, Hicks, Memon, Porter, Pugh, Shneiderman, Tseng, Zelkowitz

These three inter-related areas are covered by a single field committee and include a diverse set of activities from work on compiler optimization, to reconfiguring software on-the-fly, to the empirically evaluating software processes and products, to the development of visualization techniques. There are several faculty collaborations and research groups and laboratories, such as the Human Computer Interaction Laboratory and the Experimental Software Engineering Group.

Victor Basili's primary interest is an empirical understanding of the relationship between software processes and products. He has studied the application of techniques in software development organizations and used the data collected to build models that characterize, evaluate, predict and improve the techniques and their

effects (Goal/Question/Metric Approach, Quality Improvement Paradigm). He is involved in the development of methods and tools that support the analysis, synthesis, and feedback of project information to support organizational learning (Experience Factory). Current activities involve the development of an experience base of empirical studies, the development of stakeholder-based models for dependable systems, and the evaluation of the maturity of technologies for application in building dependable systems. He has developed families of techniques for abstracting information from software artifacts of various kinds, including requirements, OO designs, and code.

Jeff Foster's research focuses on programming languages and program analysis with applications to software engineering. His goal is to help programmers increase the safety and reliability of software while simultaneously making programs easier to write and maintain. His most recent work proposes type qualifiers as a lightweight, specification-based mechanism for improving the quality of software. Using novel, constraint-based type inference algorithms allows type qualifier systems to scale to analysis of hundreds of thousands of lines of code. As part of this research, type qualifiers have been used to find security vulnerabilities in C programs and to find deadlocks in the Linux kernel.

Michael Hicks' research is oriented primarily towards learning how to develop more flexible, reliable, and secure software. His work bridges the areas of ``systems'' and programming languages, in that he has frequently applied or developed language-based technology to solve systems problems, particularly in networking and distributed systems. His most recent research emphasis is the areas of on-the-fly software reconfigurability, programmable networking, garbage collection and memory management, and designing and implementing safe low-level programming languages.

Atif Memon's work focuses primarily on the development of techniques for state-based testing of event-driven systems such as network protocol systems, graphical user interfaces (GUIs), and web interfaces. He has developed techniques to test GUIs and used case studies to show that the GUI testing techniques are both practical and effective. He plans to conduct detailed experiments to provide further empirical evidence of their strengths and weaknesses of the techniques. To this end, he has packaged the techniques into a comprehensive tool for GUI testing, GUITAR. The tool will be available to researchers and practitioners and its deployment will provide opportunities for further research including the development of new testing paradigms for event-based systems. He is also investigating the tailoring of these techniques for protocol testing, web testing, security testing, and configuration space reduction.

Adam Porter's research interests include empirical methods for identifying and eliminating bottlenecks in industrial development processes, the experimental evaluation of fundamental software engineering hypotheses, and development of tools that demonstrably improve the software development process. The goal of a new 5-year, multi-million dollar NSF grant involving a multidisciplinary team from five universities and research institutions is to enable the dynamic analyses of software systems, around-the-worldand around-the-clock, leveraging fielded resources during local, off-peak hours. To do this the team is devising analysis techniques that are both highly distributed and lightweight from the perspective of individual system users and that are incremental and adaptive in the sense that they change their behavior over time based on earlier results. This approach is expected to give software developers unprecedented insight into the behavior of their systems as they actually run in the field.

Bill Pugh works on tools for helping make software reliable, including static tools for finding errors in object oriented designs and runtime tools that provide a "flight recorder for software". Thus when crashes or undesired behavior occurs, there is more useful information than a stack trace or a core dump. He also works

on frameworks and semantics for multithreaded, distributed and real-time middleware. This includes work on the semantics of Java threads and a pure Java message passing framework that allowed a Java implementation of the NAS CG benchmark to substantially outperform the Fortran/MPI implementation.

Chau-Wen Tseng works on software support for high-performance computing, specifically in developing compilation and run-time techniques to efficiently exploit architectural features found in modern microprocessors. His research focuses on two fundamental issues: parallelism and locality. By applying sophisticated analyses and transformations, compilers can automatically generate programs with good performance for many advanced architectures. He has developed optimization techniques in a variety of areas, ranging from reducing inter-processor communication for message-passing multiprocessors to exploiting multi-level caches on high-performance microprocessors. Improvements are demonstrated through prototype implementations and experimental results from representative programs and standard benchmark suites.

Marvin Zelkowitz is looking at the problem of understanding new software development technologies and specifically how those technologies get transferred into industrial practices. All too often new technology is promoted by hype with little empirical data supporting it. A scientific approach to technology validation often needs experimental approaches to validating this new technology. Recently completed activities includes 25 years of experiences with the NASA GSFC Software Engineering Laboratory from 1976 through 2001 as well as a recently completed study of return on investment from independent verification and validation (IV\&V) in the NASA space shuttle program. A new 5-year research project involves understanding high dependability within the NASA domain. He is also Chief Scientist of the Fraunhofer Center Maryland, where he interacts with the staff there on related empirical studies.

Human-Computer Interaction

Faculty: Bederson, Guimbretiere, Shneiderman

Ben Shneiderman works on the development and application of information visualization tools, including treemaps and starfield displays. Many of these tools have had commercial success in a variety of applications, e.g., treemaps have been applied to the stock market and business analysis, starfield displays spawned a commercial product called Spotfire. He continues to develop new algorithms for treemaps, refining the techniques of dynamic queries and extending them to new domains such as time series data. Current projects include TimeSearcher, a general purpose tool for exploration and pattern identification in time series data, and Microarray, a set of experiments used to examine changes in gene expression over time where data sets are analyzed using clusters, self-organizing maps, heat maps, and other standard microarray analysis tools. TimeSearcher is based on the use of timeboxes - rectangular, direct-manipulation queries - to support interactive exploration via dynamic queries and provides overviews of query results and drag-and-drop support for query-by-example.

Francois Guimbretiere is investigating novel interaction techniques for interactive surfaces. While his previous work focused on large vertical interactive surfaces like the white-board like Stanford Interactive Mural, his current project will explore horizontal interactive surfaces like digital tables, tablet computers and digital paper systems such as the Anoto pen. Over the next few years, he will implement and compare interfaces designed for each of these systems to understand how to narrow the gap between computers and paper.

Ben Bederson works on interaction and visualization techniques, focusing on three areas: Zoomable User Interfaces (ZUIs), mobile devices, and interfaces for children. ZUIs are dynamic contextual information displays that use spatial representations and smooth zooming for navigation. They use animated zooming to present information in context. ZUIs have been applied to tree browsing, web history navigation, presentations, and photo browsing. There are general-purpose ZUI toolkits (Jazz, Piccolo) that are broadly used. ZUIs are useful in many contexts where there is more information than fits on the screen, e.g., mobile devices with small displays. He has applied ZUIs and other techniques to PDAs for applications such as calendars, menu selection, and photo browsing. He has applied ZUIs as a base technology to a broad set of applications for children - the most recent of which is the International Children's Digital Library, a collaborative effort to provide access to thousands of books from around the world to children.

Systems and Networks

Faculty: Agrawala, Arbaugh, Bhattacharjee, Hollingsworth, Iftode,
Keleher, Shankar, Sussman

Faculty in this group are involved in many major aspects of systems and network research. Systems faculty have a well established tradition of collaborative work on projects that address system design and methodologies, system building, prototype implementations and testbeds, modeling and performance evaluation, measurements and empirical studies, and basic and fundamental studies. The current research efforts involve a variety of systems including distributed systems, pervasive systems, mobile computing, GRID computing, resource-aware computing, high performance computing, operating systems, real time systems, wireless networking, network protocols, and system security.

Ashok Agrawala and Udaya Shankar are studying primarily networking systems and network performance, including location-based systems, andhave developed systems like NetDyn and Rover. Netdyn has been used to monitor end-to-end behavior of packet losses and round trip delays and has uncovered irregular behavior of network components such as routers. As a part of project Mentor, scalable monitoring techniques based on Active Networking have been developed, along with new fast-timescale control techniques. Current results include new migration algorithms for MPLS networks, and distributed monitoring and rerouting of flows around network ``hot spots''. Rover is a software infrastructure that treats time and location as first class entities. It uses signal strength-based location-determination techniques to create radio-maps, which are used to support location-dependent computing. The base Rover protocols have been implemented and demonstrated over PDAs and laptops using 802.11 networks. A system, DRACO, based on Rover design but having the rapid deployment capabilities is being developed for support of first responders. This work involvesdeveloping techniques and architecture for location-based services and spontaneous services in mobile platforms, particularly the integration of WLAN and traditional wired networking.

Udaya Shankar is also studying performance of large networking systems involving development of efficient techniques for performance evaluation. This approach is based on the Z-iteration method, which is applicable to time-dependent queueing systems and yields time evolution of various instantaneous probabilistic measures (eg, blocking probabilities, average number of customers at resource and in service, etc) several orders faster than numerical or simulation approaches. He is developing a compositional design and analysis framework for the analysis and testing of correctness properties of concurrent systems, including real-time and security properties. The approach used is layered compositionality and assertional techniques. The testing framework is in Java and is being applied to undergraduate networking classes.

Bobby Bhattacharjee and Pete Keleher are studying multi-party security. They have developed new techniques for securing large group communications over the Internet. The work includes scalable techniques for re-keying of TerraDir, which is a distributed peer-to-peer directory protocol that can be used as the basis for implementing customized directories for Internet applications. Bobby Bhattacharjee is also working on other network security projects.
For example, NICE is a cooperative framework for scalably implementing distributed applications over the Internet. Applications in NICE are cooperative: they devote a part of their own resources to be used by any member of a cooperative group. The goal of NICE is to show that cooperative applications can achieve overall better performance than applications that do not cooperate. They have developed a set of protocols for application-layer multicast, application-layer distance estimation, and secure multicast within the NICE framework.

Jeff Hollingsworth is studying high performance computing. Two of his projects are systems called Dyninst and Active Harmony. The normal cycle of developing a program is to edit source code, compile it, and then execute the resulting binary. With Dynaist, an Application Program Interface (API) has been created to permit the insertion of code into a running program. The goal of this API is to provide a machine independent interface to permit the creation of tools and applications that use runtime code patching. With Active Harmony, a software architecture that supports distributed execution of computational objects in dynamic execution environments, with automatic application adaptation and shared data interfaces, is being studied. The unique aspect of the Active Harmony work is the emphasis on adapting to heterogeneous and changing environments. The primary result of this research will be an infrastructure and a set of algorithms that permit global resource optimization under changing conditions.

Alan Sussman is also working in the area of high performance computing. Some of his work focuses on runtime and compiler support for data-intensive applications. The goal of this research is to build a common set of software tools and infrastructure that can support the development of many classes of parallel and distributed data intensive applications. He is addressing the problem of coordinating the various components of the application, namely computation, I/O and interprocessor communication. He has been investigating these issues in both tightly coupled parallel environments and the distributed heterogeneous Computational Grid environment across multiple application domains, and has built both an object-oriented framework and a component-basedsoftware environment for creating high performance data intensive applications.
Sussman is also studying interoperability of data parallel programs. While in sequential programs applications can use simple abstractions for moving data between address spaces, such as sockets, pipes or shared memory segments, no such facilities have existed for parallel programs. He has been working on a meta-library approach to solving the problem, and has built prototype software that enables exchange of data between separate (sequential or parallel) programs, and can also be used to allow data transfers between data managed by different data parallel regions in the same program.

Bill Arbaugh is working primarily in the area of computer security. One of his projects is wireless mobility and security. In this effort the areas being investigated include fast hand-off's, probabilistic based ad-hoc routing, and ad-hoc service discovery. The research to date has resulted in several widely used software artifacts, and widely read technical reports and publications.Bill Arbaugh is also working on platform security and configuration management.
Rather than take the standard approach to securing platforms, i.e., trusted operating systems, this work is focusing on improving systems management by providing a dynamic and independent auditing capability that is OS independent.

Liviu Iftode works in the area of operating systems and distributed computing. His Split-OS is new operating system architecture for the next generation of internet servers built as clusters of intelligent devices. The lab hosts several projects related to Split-OS covering networking and file system issues as well as highly-available services. In another project called Smart Messages, he is developing a system architecture to support the computation diffusion on ad-hoc networks. He is also actively pursuing research in the area of massive networks of embedded devices such as networks of sensors.

Contents

Contents

Faculty Research

Undergraduate Program

We believe that our Department has an excellent undergraduate program. We have graduated many brilliant and well-trained students who have gone on to graduate work at top universities (MIT, Stanford, CMU, and other such universities). In the late 1990's, we grew the size of our undergraduate program to nearly 2000 CS majors to satisfy the demand from students and need by local industry. Although the sizes of our classes have increased, they are still small compared to that of some other universities, where a 150 student senior level operating systems course in not uncommon.

Curriculum

The goal of our undergraduate program is to give the student a broad based background in software and software systems. Students begin the program with a sequence of three programming courses which take them from fundamental procedural programming constructs through object-oriented programming and basic data structures such as stacks, queues, linked lists, and binary search trees. Along with these beginning programming courses, students must take Calculus I, Calculus II, and Discrete Mathematics so that they will have a firm grounding in basic mathematics. Subsequently, students take two more math courses which have Calculus II or higher as a prerequisite, one of which must be a Statistics course. At the junior level, all students must take Computer Organization, Organization of Programming Languages, and Algorithms.

It is only after completing these specified courses that students begin to have a choice of courses. Students must choose at least five courses from at least three areas. Courses offered regularly include:

Systems and Networks
Computer Architecture
Operating Systems
Computer and Network Security
Computer Networks
Information Processing
Data Structures
Database Design
Artificial Intelligence
Image Processing and Computer Vision
Graphics
Programming Languages and Software Engineering
Theory of Language Translation
Programming Language Technologies and Paradigms
Human Factors
Software Engineering
Algorithms and Theory
Design and Analysis of Algorithms

Elementary Theory of Computation
Cryptography
Numerical Analysis
Introductory Numerical Analysis
Advanced Computational Methods

We have just approved a proposal for a complete overhaul of our introductory programming courses. The new proposal is in the appendix, and the new course sequence will start in the Spring of 2004.

Advising

The goal of the advising program is for each student to have a schedule which allows him/her to be challenged but comfortable. Advising begins with the orientation sessions a student attends the summer before he/she first comes to Maryland. Students are placed in the beginning courses using a combination of Computer Science and Mathematics Advanced Placement Examinations and Department placement examinations administered in the summer orientation sessions.

Once a student is a Computer Science major at Maryland, he/she must see an advisor each semester. This policy is enforced by our registration system, which requires an advisor's electronic permission stamp for a student to register for the next semester.

The Department's Undergraduate Education Office coordinates all advising for undergraduates in Computer Science. Students who are taking 100 level Computer Science courses go to our Dean's Office for advising. In the Dean's Office there are five or so professional advisors who work closely with the students on fundamental studies requirements and choosing beginning computer science and mathematics courses. Once students reach the 200-level of Computer Science courses, they come to the Department for advising. The ten instructors for the Department's lower level courses, all of whom have at least a Master's degree, are also the advisors for the Department.

Accreditation

Like many nationally ranked computer science departments, we have chosen not to go through a formal accreditation process. The process itself takes at least a couple of years and significant effort. The curriculum for computer science changes so rapidly that the curriculum presented during the accreditation process is out of date during or soon after the process. For this reason we have chosen not to spend our effort on accreditation but on the program and its students.

Teaching Load

Instructors and lecturers teach three courses per semester and advise four hours per week. They have no research requirements. Professors teach one course per semester (graduate or undergraduate), typically teaching about 40% graduate courses and 60% undergraduate courses. We have included the teaching history of our tenure track faculty and instructors in an appendix.

Faculty Books and External Teaching Awards

Textbooks suitable for undergraduate curricula (co-)authored by our faculty include "Data Structures and

Algorithms in C++" by David Mount, and "Designing the User Interface" by Ben Shneiderman. Recognition of teaching excellence from outside the department, includes the following awards. David Mount won the Hong Kong University of Science and Technology, School of Engineering, Award for Teaching Excellence Appreciation (2001) while on sabbatical at the HKUST. Evan Golub was a Center for Teaching Excellence Lilly-CTE Fellow for 2002-2003.

Honors Program

The Computer Science Honors Program is a way to channel good students into research projects with faculty. Good students are identified early (via GPA and other means) and steered toward particular courses and particular faculty so that they can get involved in projects in their areas of interest. If they finish the research project and have maintained high GPA's, then they graduate with Honors. Several of our honors students have presented their work at conferences. Some have gone on to graduate school at Stanford, Berkeley, MIT, and other such schools. All have learned a great deal about some aspect of computer science.

In an attempt to better acknowledge the outstanding students in our department, we established the Eta Chapter at Maryland of the Upsilon Pi Epsilon International Honor Society for the Computing Sciences in the summer of 2001. This international honor society was founded at Texas A&M University in 1967 and has since grown to include over 150 institutions.
The dozen or so students selected annually for nomination in the Computer Science Department at our campus have shown great talent in their academic pursuits towards their degree and have merited invitation by achieving a superior grade point average in their Computer Science, Mathematics and Statistics courses while at Maryland

Number of Majors

The table below shows the number of students official listed as majors in our program for the past five years.

Year	1998	1999	2000	2001	2002
Number of majors	1752	1995	1939	1895	1665

The table below shows the number of incoming Computer Science freshmen.

Year	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002
# Freshmen	102	106	170	174	191	314	297	321	349	326	177

In fall 1997 the number of incoming Computer Science freshmen increased significantly and stayed at that level for about three years. For the last two years, the number of students entering the program has decreased. Through all of the fluctuations, we have managed to accommodate the needs of our students through the hiring of additional faculty and instructors, and through increased class size (the number of students in most senior level courses was increased from 40 to 50 students). It is very rare for a student to get closed out of a class. When this does happen, it is an upper level requirement, which can be met with other classes, so we place the student into an alternative class, which moves him/her toward graduation.

Gemstone Program

Gemstone is a unique program on our campus that involves honors undergraduate students of all majors. These students form teams of up to 15 students during their freshman year, and each team focuses on a specific research topic that culminates in a formal written thesis during the senior year. Each project, guided by a faculty mentor, must not only incorporate technological research, but also must involve social relevance. Current Computer Science Department faculty serving as team mentors, and the topic areas of their student teams, are:

- Bobby Bhattacharjee: Network Security;
- Don Perlis: Universal Device Interface;
- Jim Reggia: Genetic Programming.

Further information on this program can be found at http://www.gemstone.umd.edu/

Retention

In the six years from fall 1994 to fall 2000, the freshman retention rate for Computer Science majors rose over nine percent from 67.1% to 77.9%. It rose from 73.2% to 77.9% from fall 1997 to fall 2000. This rise is due in part to a focused concern on the part of the Department and the College. The Department instituted an Undergraduate Teaching Assistant Program. Undergraduates who have excelled in lower level programming courses and in the discrete structures course are hired to provide additional office hours for these courses. Many students have been helped by this additional free tutoring which is conducted by students who are especially trained and informed on the particular course assignments. The Department also made a prerequisite change two years ago which seems to be having a positive effect on success rate in the lower level Computer Science courses. Many aspiring Computer Science students were weak in mathematics and did not seem to understand the importance of mathematics in a Computer Science program. In an effort to let students know early about the mathematical content of the program and to improve the retention rate, the Department changed the corequisite for the first programming course to Calculus I. (Calculus I had been the corequisite for the second programming course.) In addition, the prerequisite for the discrete structures course was changed to Calculus II (rather than a co-requisite of Calculus I). This change has vastly improved the success rate for the discrete structures course.

The Dean's Office has a couple of programs which identify and aid at-risk students. In one of these, Project REAP, the student meets with a small group and a trained instructor each week. The discussions center on approaches to being a successful student. Concrete exercises are a part of the course. It is also very important to note that our mandatory and careful advising is a major factor in retention.

Number of graduates

The table below shows the number of undergraduate degrees granted during each of the past five years:

Year	1998	1999	2000	2001	2002
Graduates	182	194	222	282	269

Placement of graduates

Though it is difficult to gather placement information on graduates, all indications are that the placement for

our graduates is nearly one hundred percent. Our Dean's Office does a survey of students who are graduating in a given semester. In spring 2001, for example, forty four students indicated that they had accepted jobs. Some of the companies which hired more than one student were Microsoft, The Motley Fool, Hughes Network Systems, Lockheed Martin Corporation, and the National Security Agency. The average starting salary was a bit over $52,000. Three of the Computer Science graduates indicated that they were going to graduate school. Several had not begun to look for jobs. Others said that they were deciding between multiple offers. We do not have more recent compiled results of these surveys.

Feedback from students

We spoke to a sample of our undergraduate students. They were happy overall with our undergraduate program. The main issues that they brought up are as follows.

1. Lack of real contact with faculty, due to the large size of the department. The students felt that they had little opportunity to get to know any of our faculty. One student contrasted his 50-student senior-level courses in our department with his 15-student courses in Philosophy.
2. Approaching, finding and waiting for a TAs for one's class in the large TA room is chaotic and can be intimidating for new students.
3. Students want human feedback on programming projects for issues such as style, etc. They found that some number of students even at the Senior level were seriously lacking in overall programming skills, and felt that regular feedback during course projects would help substantially. This is currently a logistical issue, since we have a small number of instructors and faculty relative to the number of students.
4. We need better labs—many have old/slow computers. Some students need Windows machines. The Business School building has better computers, but is too far away. The Linux Lab closes by 10PM, but many students prefer to work late on programming assignments.
5. Wait-list roulette is a continuing source of frustration for the students. A particular problem exists for our best sophomores and juniors, who may be ready to take senior level courses but are shut out of them by the time they are allowed to register. It was suggested that a few seminar-type senior courses with a small cap will help the more motivated students. We also need to find ways for talented undergraduates to take (advanced) graduate courses. In general, we need to continue working on the balance between serving all students and focusing on both the stars and the students lagging behind.
6. Some of our best students are either ignorant of or avoid the honors program. Perhaps some avoid it because they cannot commit the time to take on an honors project. A few seminar-type courses, as mentioned above, can provide a challenge for interested undergraduates.
7. General problems caused by lack of community among the undergraduates.

We also reviewed comments from several students that felt that the environment within the department wasn't supportive and that the programming assignments were excessively difficult and time consuming.

Feedback from instructors

We also held discussions with the instructors. Among the many issues discussed, we settled on the following three goals as important goals for the next several years.

- Getting back to 40-student caps in the 400-level courses.
- Providing some human feedback on programming projects in all courses with such projects.

- Doing more to provide review and professional development for instructors. One goal of instituting a different review process is to strengthen the communication between instructors and professors. Avenues for professional development include getting one course off periodically to take courses; opportunities to participate in research proposals; equivalents of Summer Research Grants, etc.

Recommendations for our Undergraduate Program

The state of our undergraduate program is good, but not great. Attempts are being made to make our undergraduate program world class, but resource constraints are limiting that effort.

A surge in undergraduate enrollment, starting in 1997 and unlimited by any selective enrollment or admissions caps, more than doubled the size of the undergraduate program. The increase was seen both in the number of freshmen electing to major in computer science, and in the number of students in each course. A new undergraduate major in Computer Engineering was also started in 1997 in the Department of Electrical and Computer Engineering, and those students are taking several of our classes. Some junior level courses more than tripled their enrollment, and the student surge recently reached the senior classes.

To cope with this surge in enrollment, the CS department hired a number of additional professors and instructors, but was unable to hire enough to balance ith the increased number of students. As a result, we agreed to a number of temporary measures, such as increasing the size of our senior level undergraduate classes, leading to a higher student to faculty and TA ratio.

The number of full time lecturers has substantially increased over the past 5-10 years. The lecturers are considered faculty, and participate in faculty meetings and last year's departmental retreat. However, professors and lecturers have not co-taught courses, and we still need to do a better job of building communications between the instructional and the professorial faculty.

While we were able to maintain adequate quality in our core undergraduate mission, other tasks associated with a world-class undergraduate program were given less energy, or have simply been impossible. These tasks include:

- Providing students with substantial feedback on style and good programming habits on their programming projects (in addition to grading on correctness). Our current student/TA ratio is 33-50 students/TA, which doesn't allow for much feedback.
- Encouraging informal, out of class contacts between students and faculty.
- Small senior capstone and/or honors courses.
- Involvement of professors in the lower level programming courses

The number of incoming CS freshmen is dropping back down to pre-boom numbers, and the number of students in the lower level courses is dropping. There is significant agreement and enthusiasm in the department for trying to improve our undergraduate program and taking on some of these neglected tasks, including a complete revision of the lower level courses and having professors involved in the development and teaching of those courses.

The space and administrative staff is also woefully inadequate even for the undergraduate program. The space available for advising is completely inadequate, and there is no space for an undergraduate lounge or undergraduate research laboratories. Our undergraduate office is understaffed, with inadequate manpower for advising, tutoring and working with students on extracurricular activities.

We do not have enough information to know if the complaints by some students about our program are

representative or atypical. Over the past few years, since we had no limited enrollment and a huge number of majors, many of the lower level courses had relatively high failure rates. This may have contributed to an environment that was perceived as unsupportive. We should do more to investigate this issue, and do whatever we can to make the undergraduate program more supportive, particularly to bright students that do not have substantial programming experience (a category that includes many of our female students).

While dropping student numbers might have freed up the resources needed to take on these tasks, the current budget crunch has forced staffing reductions that threaten our ability to substantially improve our undergraduate program. For FY 04, the campus budget cuts have forced the elimination of 3 lecturer positions and 12 graduate TA positions. There has been a freeze on the hiring of new faculty. A bleak budget situation is likely to extend for at least another year, perhaps longer.

In spite of these reductions in staffing, lower student numbers will allow us to maintain an adequate but strained level of quality in our core undergraduate mission. However, we will be forced to delay important enhancements to our undergraduate program.

The critical issues that need to be addressed are:

- The size of most of the senior-level courses should be reduced to at most 40 students, in line with most other majors on campus.
- TA support in the lower level programming courses needs be increased to allow providing students with feedback on good programming habits.
- We need to find and allocate the resources needed to continue the planned revision of the intro programming courses and involvement of professorial faculty in the development and teaching of those courses.
- These changes should be accomplished while not noticeably decreasing the percentage of graduate courses taught by professorial faculty.
- We need to establish a better practice of reviewing, mentoring and providing professional development opportunities to the instructional faculty. This effort should involve the professorial faculty, to help improve communications between the professorial and instructional faculty.
- We need additional administrative and instructional staff to provide better customer support for our undergraduates.
- We need to evaluate what steps, if any, need to be taken to make the students feel that the department is a more supportive environment, particularly for those students without extensive programming experience before their major.
- We need more space for TA office hours, for the additional administrative staff needed by the undergraduate program, for an undergraduate lounge and for undergraduate research laboratories.
- We mist provide more opportunities and encouragement for our best students to have informal contact with faculty and to get involved in research.

Graduate Program

Contents

Contents

Previous, Undergraduate Program

Graduate Program

The goal of our graduate program is to provide our students with the necessary knowledge and skills to achieve successful careers in research in academia and industry. In order to achieve this goal, the Department has endeavored to

- recruit highly qualified students of diverse backgrounds
- provide these students with a diverse curriculum and program structure that encourages both a broad understanding of Computer Science as well as depth of knowledge in their areas of specialization
- provide training, mentoring and guidance for students in the skills needed to teach and conduct research
- provide students with the necessary resources to achieve these goals, including financial support, access to computational facilities, adequate office space and equipment
- provide a supportive, stimulating and diverse intellectual environment
- provide clear and regular assessment of their progress

In this section, we will provide a factual overview of the graduate program, and present the ways in which the Department has attempted to achieve our goal of quality graduate education. Later, we will present an assessment the strengths and weaknesses of the graduate program, and suggest some ideas and possible directions for improvement.

Admissions and Student Body

Our graduate program enrolls roughly 200 full time graduate students. The majority of these students are Ph.D. students. (The department does not really distinguish between M.S. and Ph.D. students. Virtually all graduate students are treated as potential Ph.D. students, even though a number leave with their M.S. degrees.) Over recent years we graduate on average around 25 M.S. degrees per year and around 18 Ph.D.'s per year.

We receive roughly 500-1000 applications per year. Each year we make around 150 offers of which roughly 30-40% are accepted. We have observed a recent increase in the number of applicants from 400 to more than 1200 over the last four years. This increase is due in part to the fact that the economic downturn makes graduate school more attractive.

We feel that the overall quality of the graduate students is very high. Applicants who refuse our offer of admission typically go to stronger departments, such as Berkeley, CMU, Illinois, and Wisconsin. The following table shows recent trends in graduate admissions.

Summary of Graduate Admissions								
Year	Offers		GRE			GPA	Acceptances	
	Total	Int'l	V	Q	A		Total	Int'l

1999	--	--	553	781	724	3.5	30	19
2000	160	109	535	788	730	3.6	64	55
2001	120	81	543	775	728	3.6	38	27
2002	127	63	578	775	761	3.6	47	30

Currently we have 206 full-time and 34 part-time students. Of these around 15-20% are women, 60-65% are international students. Around 3-5% of our US students are from underrepresented minorities (African Americans, Hispanics, and Native Americans).

For students entering with a B.S. degree, the average time to graduation with an M.S. degree is around 2 to 2.5 years and, the average time to obtain a Ph.D. ranges from 4.5 to 6 years.

Support

Students who are admitted with aid are guaranteed funding for two years, but as a practical matter, no full-time student in good standing goes unfunded. Students who are working under advisors in poorly funded areas are usually provided with teaching assistantships to make up for any lack of funding. Among full-time students, roughly 60% are funded a graduate research assistants, 10% are funded (totally or partially) by fellowships, and around 30% are funded as teaching assistants. Graduate stipends are listed below.

Graduate Stipends			
	Teaching Assistants	Research Assistants	
	9.5 Month	9.5 Month	12 Month
Step I	$14,522	$15,081	$21,030
Step II	$14,801	$15,639	$21,809
Step III	$15,081	$16,198	$22,588

The vast majority of teaching assistants are paid at the Step I and II levels. Most work for 9.5 months, but roughly a quarter are employed over the summer, where they may earn an additional amount per course ranging from $2500 to $3000.

In a study from the University of Pennsylvania on graduate stipends from 2000-2001 at the 20 top computer science departments, Maryland was ranked in the lowest third. Even though stipend levels have increased marginally over recent years, our stipend levels have lagged behind many of our peer departments. In order to make our admission offers more competitive, we have a practice of distributing our University fellowship funds over many offers in order to provide “fellowship bonuses” to a number of the stronger applicants. Nonetheless, given the relatively high cost of housing in the area combined with poor management of graduate housing on campus, our low stipend levels continue to be a source of dissatisfaction with our graduate students.

Program Organization

Our graduate program strives to provide students with a breadth of understanding of computer science, while encouraging their early entry into research. We will focus here on discussion of the Ph.D. program, as the requirements for a M.S. (without thesis) is largely a subset of these.

Courses: Courses offered in the department are organized around seven different areas (with some courses overlapping multiple areas).

Artificial Intelligence
Computer Systems
Database Systems
Software Engineering/Programming Languages
Scientific Computing
Algorithms and Computation Theory
Visual and Geometric Computing

Our 400-level courses are primarily senior undergraduate courses, which can also count for graduate credit. They are offered every semester. The intention is that graduate students should take them only to acquire needed background. The 600-700 level courses form the core, long-standing graduate curriculum. They are offered on a rotating basis, each roughly every three or four semesters.

Since computer science is a rapidly changing field, we use 800-level advanced topics courses as a mechanism for introducing new graduate courses into our curriculum. These courses are offered in many forms, including pure seminar structure, pure lecture structure, and various combinations of these. Often these courses are offered multiple times, and may subsequently become part of the permanent curriculum. Some of these courses have been quite successful. For example, CMSC 818L, Network Centric Systems by Prof. Liviu Iftode was featured in the educational column of IEEE Pervasive Computing magazine.

Because of the size and diversity of our faculty, we have been able to offer quite a number of special topics courses on a regular basis. For example, in the three-year period from Fall 2000 to Spring 2003 we offered 35 such courses, or roughly six per semester. By in large, the topics of these courses reflect the research interests of the faculty. There are some areas where we would like to offer more such courses. For example, because of the recent loss of two faculty members in the database area and the need for the remaining database faculty to cover the core database courses, we have not been able to offer as many advanced database courses as we would like.

Ph.D. Qualifying Courses: In order to prepare students for their transition into research, the department has a course-based system for providing students with the necessary breadth of knowledge of computer science. Our requirements have changed very recently. Under the existing system, sometime within their first five semesters, each Ph.D. student must complete a 10-course sequence with two courses in each of five different areas. At least one the two courses in any area must be at the 600 level or higher. Students are required to get an A in at least 7 of the 10 courses and no less than a B in the remaining courses. Courses from other departments may be applied after approval from the graduate director.

Although this system achieved the desired goal of providing breadth, a number of deficiencies have been observed over time. For example, the requirement of taking 10 courses was seen as a burden to students entering the program with stronger backgrounds, since they may have already completed many of these courses at other universities. Further, the grade requirement tended to encourage students to focus exclusively on their course work for up to five semesters, thus delaying their entry to research. It also encouraged students to pad their course selection with 400-level courses in order to get more A’s.

For these reasons, the faculty has recently approved a proposal to amend the Ph.D. qualifying course requirements as follows.

Within the first five semesters, each Ph.D. student shall take 7 graduate courses at the 600 level or higher, spread over at least 5 areas, with no more than 2 courses in any one area. (See Graduate Course Offerings for courses and areas.) Students must obtain at least 5 A's and no less than a B in the remaining courses. A student may take a 400-level course for background, but this course will not count towards the core course requirements.

To provide depth, students are required to take two additional courses at the 600 level or higher, with the approval of their advisor. They must receive a grade of B or higher. There is no time limit on taking these courses. The courses may be seminar courses including those offered outside of the CS Department.

The new system reduces both the breadth and time requirements somewhat, while encouraging students to focus on graduate-level courses and courses within their area of specialization. There will also be a requirement that new students take a 1-credit course that will introduce them to various approaches and issues that arise in doing research.

Preliminary Exam and Defense: After successfully completing the qualifying coursework, the next step is the Ph.D. Preliminary Examination. This is an oral examination to review and appraise a student's proposed dissertation research, to test how well the student is prepared for the research. The student in conjunction with his or her advisor prepares a dissertation proposal that describes the proposed research, surveys relevant literature, and includes reading lists for three areas of knowledge related to the proposal. The preliminary exam consists of a student's presentation of the proposal, and an oral examination on the content of the proposal and the student's three reading lists. After passing the preliminary exam, the student proceeds to work on his or her Ph.D. dissertation. The Ph.D. is granted on the successful completion and oral defense of a Ph.D. dissertation.

Advising

Generally, we feel that there are enough faculty members to meet the advising needs and interests of our students. One notable exception is that we have no one in computer architecture, but students can work with affiliate faculty from the ECE department in this area. All new students are assigned an initial advisor when admitted and meet with this advisor prior to starting their course work. Students may remain with this advisor or switch to accommodate their research interests.

Placement of graduates

The department has not kept complete and detailed records of placement of our PhD graduates. However, from the incomplete information currently available to us, we would estimate that our PhD graduates during the two most recent years (2001, 2002) took positions mostly in academics or industry (roughly in equal amounts), with a minority accepting research positions (e.g., ARL, Lincoln Labs, postdoctoral positions). Students graduating during this period took faculty positions at Washington University (St. Louis), Rutgers University, Lehigh University, Brown University, Arizona State University, University of Louisiana, University of Pittsburgh and several foreign universities.

Evaluation of the Graduate Program

Although the department feels that, on the whole, our program does a very good job in preparing students for careers in academia and industry, we continually strive to improve the quality of the program and tune it to the needs of our graduate students.

In order to better gauge the effectiveness of our program, we asked all of the graduate students in the program to participate in a survey. The survey consisted of around 75 questions, including both multiple choice and written responses. A summary of the results of the multiple-choice questions can be found in an appendix. We received 91 responses to the survey. Below, we summarize a number of the findings of this survey.

Courses and Curriculum: Overall there is wide agreement that our curriculum is very diverse and provides students with valuable knowledge for their future career goals. Some survey respondents expressed a desire to see more course offerings in some areas, including computer architecture and databases. Although the faculty agrees with this assessment, our ability to offer courses in these areas has been hampered either by a lack of or recent loss of faculty in these areas. Some respondents also expressed a feeling that there is too heavy an emphasis on course work, and that alternatives leading towards research (projects or research papers) should be considered.

The survey respondents generally felt that the current 10-course Ph.D. requirements are reasonable in terms of the number and breadth of courses. There was less agreement about the time and grade requirements. A number of respondents expressed a desire to have fewer courses or to provide students with greater flexibility in choosing the areas of their courses. A number of respondents noted that the need to focus on courses in order to satisfy the grade requirements impeded their entry into research. The faculty has acknowledged this sentiment, and we hope that the new Ph.D. course requirements will address many of these concerns.

Teaching Experience: There were a number of criticisms of the experience that graduate students gained as teaching assistants. On the positive side, survey respondents felt that workloads were reasonable, and many felt that their teaching experience was quite valuable. However, many expressed the feeling that the Department could do a better job in providing students with training and guidance in teaching. Although it was felt that some course supervisors do a very good job in providing guidance and direction to their teaching assistants, many respondents would like to see much more of this. There was also sense that a better job could be done in assigning students to courses based on their particular backgrounds and interests. A number of respondents commented that they would like to have more opportunities to teach. There was also some concern about the cramped space used that is currently being used for TA office hours. (The Department is working on equipping a larger room for this purpose, which it hopes to have available in the fall.)

Facilities, Travel and Administrative Support: There were widely expressed concerns among the survey respondents regarding the adequacy of building and office space. For the most part, graduate students are housed in large interior offices in the A.V. Williams building. These offices are without windows and can be quite crowded. A few respondents complained of poor ventilation and erratic air conditioning in the building.

The general stress on the availability of space for student, staff, and faculty offices has resulted in the recent closure of social space, including our graduate lounge. Currently, the faculty, staff and students share a small common lounge area. This has the regrettable effect of reducing the quality of social interaction among all these constituencies. We also have no good space for group study for graduate students. On the positive side, the recent construction of the adjoining CSIC building has greatly enhanced the quality of our instructional facilities, and provides some additional meeting space for faculty and students.

One of the consequences of the University's recent budget cuts has been the need to curtail our program of Departmental support for student travel. A number of respondents noted that this has adversely limited their ability to select conferences to which they will submit papers.

Respondents were nearly unanimous in their praise of the efforts of our administrative staff. The general feeling is that our staff works tirelessly, but under great stress. Many respondents acknowledged that being understaffed adversely affects the timeliness with which their requests can be processed. Respondents were also relatively happy with the quality of access to computer equipment and library facilities.

Advising, Guidance and Placement: Overall, the survey respondents indicated strong satisfaction with their advisor and their research. However, the quality of training in career skills, including public speaking and writing research papers, is very much dependent on one's individual situation and advisor. A number of respondents expressed a desire to see more institutional support for this type of training. For example, from time to time, the Department has offered informal seminars on topics related to job skills. (One example is a recent seminar by Profs. Lise Getoor and Amitabh Varshney on finding academic positions.) A number of respondents expressed a desire to have more of these.

We also asked students about evaluation of their progress. Although the vast majority of students proceed smoothly through the system, there are a small number of students who after some number of years in the system have failed to find an advisor and seem to be drifting without direction. Currently we have no system in place for the regular evaluation of student progress. Some respondents commented on the fact that students receive little evaluation of their progress, except when a deadline is about to pass. They also comment that the task of finding an advisor can be quite intimidating, and that it would be nice to have better ways for faculty members to establish connections with students in search of advisors. In recognition of the importance of providing students with regular feedback, the department has formed a committee to study this issue and to provide a recommendation for a system that will check on the progress of students on an annual basis, and provide feedback to students that are in need of guidance.

Climate and Environment: In terms of their course work, research, advising, survey respondents were generally satisfied with overall quality of their experience in the department. They felt that the department faculty is supportive of their needs, and that there is a supportive community of students. A significantly number of respondents expressed a sentiment that they wished that they had more time to pursue interests outside their academic program.

A significant number of survey respondents also expressed concern about the level of funding, especially in its relationship to the cost of housing in the area. There was wide agreement that it is not possible to find adequate housing in safe neighborhoods given current funding levels. Although it is not clear what steps the Department can take to improve matters here, this seems to be an issue of great concern to our graduate students (and probably graduate students of other departments), and providing sufficient financial support so that our graduate students can obtain adequate housing is an important institutional goal for the University.

Next, Administrative Infrastructure

Contents

Contents

Previous, Graduate Program

Administrative and Physical Infrastructure

Administration and Budgets

Administration

The Chair is directly supported by two positions.

1. Pat Ipavich is the Director of Administration and manages the Director of Business and Finance, the department's Librarian/Information Specialist, the Chair's Administrative Assistant, and a Business Services Specialist responsible for keys, phones, parking assignments, furniture, building maintenance and mail.
2. Adelaide Findlay provides secretarial services to the Chair.

The Computer Laboratory staff is directed by Brad Plecs, who is currently supported by five full time professionals. They are responsible for both the research and administrative computing facilities of the Department. With the exception of a few specialized laboratories, computing for undergraduates is provided by the campus's Office of Information Technology.

The undergraduate office, directed by James Maybury (who also has instructional responsibilities) is responsible principally for undergraduate advising, and management of teaching assistants and instructional staff. Two faculty members – Profs. William Gasarch and James Purtilo – share the responsibilities of faculty undergraduate directors. Mr. Maybury is supported by one administrative assistant.

The graduate office is directed by Gwen Kaye. That office handles admissions, schedules all classes (undergraduate and graduate), and makes TA assignments, maintains records of all current graduate students, and conducts orientation sessions for new students. Ms. Kaye is supported by one administrative assistant.

Staff support for all activities is one of the greatest challenges facing the Department. While the Chair's office was supported by another staff member responsible for public relations and outreach, budget reductions have forced us to eliminate that position. The person in this position was going to help us design and implement a campaign to raise funds for a chaired position in our Department. We felt that this was critical to help us recruit at the senior level. Now, we will need more support from both the College and the Campus to make a CS chair a high priority funding item. Finally, when the previous assistant who was responsible for facilities and mail resigned, we promoted an Executive Administrative Assistant from the chair's office to that position and left the chair's executive assistant position unfilled. The sections of the report on graduate and undergraduate education discuss the impact of staff shortages on those operations.

The table below shows staff support for these three functions – chair's office, undergraduate education and graduate education – at three of our aspirational peers – University of Texas at Austin, University of Washington and University of Illinois - and compares them with staffing at Maryland. In all cases, our

department is understaffed in comparison, and this has had a substantial impact on our ability to support our programs and students.

University	Chair's office	Undergraduate office	Graduate Office
Maryland	2	2	2
Illinois	5	4	4
Texas	3	3	3
Washington	4	3	3

Budgets

There are several sources of support for the Department's salary and operating budget. These include:

1. State budget – funds provided by the campus through appropriations from the State legislature.
2. Access – a program on campus that channels funds, on a temporary basis, to departments with significant stress on their undergraduate education programs.
3. Summer school – which returns funds to academic departments proportional to tuition raised from their summer offerings.
4. DRIF – which provides funds to Departments based on contract and grant overhead generated in previous years.

In addition, between 1999-2003 there was a fifth significant source of funds – the Maryland Applied Information Technology Initiative (MAITI) – which was a state initiative to increase the number of graduates in the information technology area. This program terminates during this fiscal year. When the program was initiated, it was intended that the funds temporarily provided by the program would be replaced by permanent base budget increases to the State budgets of participating institutions and departments.

The table below shows the Department's state budget for the past eleven years:

Year	Budget (millions)
1993	3.06
1994	2.88
1995	3.15
1996	3.10
1997	3.23
1998	3.61
1999	4.23
2000	4.64
2001	5.19
2002	5.80
2003	5.56[1]

The decline in funding from FY 2002 to FY 2003 is a combined result of a rescission to our State budget of 5% (approximately $270,000), and a decline in our MAITI funding of $450,000, which was partially offset by an increase to our base budget of $275,000 (see below). Funds from ACCESS, summer programs and DRIF have remained steady at approximately $325,000, $60,000 and $150,000 annually and respectively. Funds from MAITI, however, have dropped precipitously during the past two years. That funding profile

was:

Fiscal Year	MAITI funds
1999	200,000
2000	150,000
2001	640,000
2002	601,000
2003	153,000

The decrease in funding from 2002 to 2003 was very significant – $450,000. This was partially compensated for by a permanent increase to the Department's state budget of $275,000 provided by the Provost. But the remaining MAITI funds disappear altogether next year, so that the Department's revenues will decrease by another $150,000 from FY 2003 to FY 2004.

While budgets are decreasing, the Department's faculty salary obligations significantly increase next year because two faculty members hired last year come on board at 100% only next year. Associated with this are increased, although temporary, costs for startup packages for those faculty.

The bottom line is that if the Department were to operate next fiscal year (2004) at its current rates of expenditures, it would end that year with a deficit of more than $800,000. How have we dealt with this? The following steps have been taken to decrease spending, on the one hand, and increase funding, on the other:

1. Three full time instructors will be released at the end of this academic year, resulting in a saving of $150,000. With current declining enrollments the Department will still be able to provide enough sections of courses at introductory levels at the same level of staffing as this year. Unfortunately, we had hoped that the declining enrollments would allow us to use our resources to significantly improve the undergraduate experience for our younger students, but at this point we will only be able to maintain the current level of quality. This is discussed in more detail in the section on undergraduate education.
2. Teaching assistants will be reduced by twelve, saving approximately $180,000.
3. Operating costs, mostly associated with the department's computer support laboratory, will be reduced by $50,000.
4. The two staff positions in the chair's office will remain unfilled—$70,000
5. The Dean is providing the Department with an additional $100,000 in permanent base budget.
6. The Provost is providing the department with one time funds of $280,000 to ensure that we have sufficient instruction and TA support in our upper division courses.

These measures will allow the Department to end next year with a balanced budget, but provide no funding for needed staff growth, faculty recruiting or improvements to our undergraduate or graduate programs.

Physical Infrastructure

Most of our Department's office and lab space is in the A.V. Williams Building. An appendix provides a detailed tabulation of how this roughly 37,000 square feet of space is used for offices, laboratories, conference rooms, library, etc. Growth in our research and education programs over the past decade has resulted in significant stress with respect to space. In particular:

- Research assistant offices were subjected to a process of "densing up" over the past four years – at

significant expense we renovated RA space so that we could offer every supported graduate student a desk and a workstation. As a consequence, these large internal offices, which should ideally also serve as labs for research groups, have no space for demonstration areas or meetings.
- There is almost no informal meeting space. We have only one very small lounge which is shared by faculty, graduate students and staff. Conference room space is also at a premium, although we have turned one internal office into a conference room to partially alleviate this problem.
- Until this year, there was almost no space for teaching assistants to meet with undergraduates during office hours (the TA offices have also been "densed up" and cannot be used for this purpose). Next year, at the expense of closing our only undergraduate PC computing laboratory, we will be adding another 600 square feet for TA advising.
- There is no more laboratory space to support any new research activities. While we can probably absorb more graduate students into the grad student offices through additional "densing" actions, there is just no solution in hand for supporting new experimental activities.

While many of our aspirational peer institutions – especially Texas, Illinois, and Washington – have made significant investments in physical infrastructure for computer science research, Maryland has not been able to do this yet. A new computer science research facility is on the campus's long range building plan, but it could realistically be more than a decade before plans are drawn up, let alone ground being broken. Our current building was financed internally by the campus, and put up quickly. It essentially has no architecture, and was certainly not designed to support computer science research and education. It might be our greatest impediment to improving our national ranking. Until such time as the campus makes raising funds for a new computer science building one of its very highest priorities, we see little prospect for the situation being improved in any significant way.

The campus has, on the other hand, provided our department with excellent space for education. This past year, the new Computer Science Instructional Center (CSIC) opened. The CSIC is located adjacent to A.V. Williams, and is connected to it by a sky bridge. It contains a 140-seat lecture hall, two 90-seat lecture halls, seven 35-to-50-seat classrooms, a Linux instructional lab, and support space. All classrooms have built-in video projectors, computers, laptop connections, and wireless Internet access. The large lecture hall is equipped with video cameras to allow lectures and classes to be both broadcast over the Internet and recorded for later playback. A small, student-staffed technical support room, a control room, a pantry, and a lobby provide for conferences and gatherings. The control room provides for full projection and recording visual and voice - as well as the transmission of internet and web services - for the three lecture halls.

Computing Facilities

The Department maintains its own computing facilities in support of both its research and graduate educational missions. While most computing resources for undergraduate class work are provided by the campus Computer Science Center, the department's own computing resources support some advanced undergraduate courses with special needs. Six permanent staff members maintain the Department equipment: a Director of Computing (Brad Plecs) and five Systems Programmers. Six to eight undergraduate students work with the full time staff providing support services.

Each Department office in the A.V. Williams building has one or more wall plates, which contain ethernet, fiber optic, and telephone outlets. Ethernet and fiber outlets are connected to ethernet switches running at 100 Mbit and Gigabit ethernet speeds, and running on a gigabit ethernet backbone. Cisco routers connect the

building switches to the campus network and the internet. The Department has a wireless ethernet network as well, allowing mobile computing users to remain connected to the network while in meetings, conference rooms, hallways, visiting other offices, or roaming certain parts of the University of Maryland campus.

Current research facilities include over 300 workstations running Sun Solaris, 200 PCs running Microsoft Windows, 50 PCs running Linux, and about 30 Apple Macintoshes, half running the new OSX and half running the older OS9. Six public laser postscript printers, a color scanner, and a color laser printer are available for use.

Most faculty have one or two workstations in their offices. Graduate students, approximately 8 to 16 in each laboratory of 500 to 800 square feet, have at least one workstation per person in each room. There is one public workstation lab with 16 workstations for the use of graduate students, and about 100 workstations around the department that are part of a public cluster. The labs are used for research, class work and special projects by both undergraduate and graduate students. Two instructional labs are available for undergraduate classes, one with 20 Windows 2000 PCs (to be closed at the end of this academic year, as discussed above), and one with 25 Linux PCs.

The Department administrative staff - the Chair's Office, Business Office, Undergraduate and Graduate Offices - also depend on the Department computer facilities. Administrative staff use primarily Microsoft Windows machines and some Apple Macintoshes.

Department Library

The Computer Science Library has a collection of about 2600 books, 202 journals, 62 of which are currently received, and over 5000 technical reports. The library also provides course reserve readings and reference books. In addition, the Department's librarian provides reference assistance to the Department and to UMIACS. Reference assistance ranges from simple informational and directional questions to more intense research questions. In 2002, the library answered 4679 basic reference questions and 565 research reference questions. The librarian also keeps the Department and UMIACS informed on the University Libraries' new sources and programs. The electronic resources and databases have been of great interest to library users including IEEE and ACM publications. The librarian is responsible for the publication and maintenance of the Technical Report collection, both paper and electronic. Due to budget constraints, there was a need to combine staff responsibilities. The librarian, whose title was changed to Information Specialist, is also responsible for the coordination of our High School Programming Contest and the department's newsletter.

Engineering and Physical Science Library (EPSL)

The very large EPSL (located two blocks from our Department) and other University Libraries are additional resources for our faculty and students. While the CS Library focuses on providing materials for the Department's graduate level courses, the Engineering Library focuses more on Electrical Engineering materials. Furthermore, the Computer Science Librarian of the EPSL consults our Department's librarian and a rotating faculty liaison with regard to collection management, especially the journal collection of EPSL. The journal collection at EPSL is much larger that ours, and through planning over the last several years, there is little repetition in the EPSL and Department. The University Libraries have recently provided access to the IEEE electronic library, IEEE Xplore, to the ACM Digital Library, and to numerous online journals and full-text databases. These electronic resources allow for easy access to important research materials.

Diversity

An appendix tabulates the gender and race composition of faculty, staff and students in our Department. The Department has actively recruited women faculty members, having made offers to many over the past five years and hiring two (Lise Getoor and Leila DeFloriani, who will be joining the department this summer). Startup packages for female faculty candidates include maternity leave benefits over and above those offered by the University.

In November 2001, 20 women and minority graduate students from the US and Canada came to our campus for a workshop entitled "Research, Careers, and Computer Science: A Maryland Symposium." The participants presented their research, met Maryland faculty and students, and participated in interactive sessions about academic jobs. These sessions focused on teaching, research, obtaining funding, the tenure process, distinctions between academic and other jobs, and time management. This event was sponsored by both the Computer Science Department and UMIACS. More information can be found on the workshop website: http://www/cs.umd.edu/users/oleary/workshop/.

In addition, approximately five years ago, women students in the Department of Computer Science started a Society of Women in Computer Science. Since that time this group has become a student chapter of Association of Women in Computing. The purpose of the organization is to promote the education of women in computing, to further the professional development and advancement of women in computing, and to promote communication among women in computing. The group is very active in the Department with a Department wide book swap at the beginning of every semester, mid-semester sponsorship of information sessions on senior level courses to be offered in the subsequent semester, women graduates roundtable on transition from undergraduate days to the workplace or to graduate school, corporate presentations on on-going projects, and an end-of-semester picnic with the student organizations Association of Computing Machinery and S.C.O.R.E.

Contents

---

[1] At the beginning of this fiscal year the Department's State budget was $5.92M, but mid year reductions decreased that budget to $5.56M.

Contents

Plan of Organization

Department of Computer Science
University of Maryland
College Park, Maryland

September, 2000

Purpose

The purpose of the Plan of Organization is to specify the means by which the Department of Computer Science (the Department) carries out its educational and research responsibilities and plays its proper role in University affairs. This Plan supersedes all previous versions.

Article 1. Department Chairperson

The chief administrative officer of the Department is the Department Chairperson.

1. Appointment. The Chairperson is appointed by the College Dean (and ultimately by the University) to a fixed term of office, usually 5 years. Prior to a new appointment or the reappointment of a chairperson, expressions of opinion of the Department Council and of the Department faculty shall be given to the University administration. No person shall serve as Chairperson for more than 10 consecutive years.
2. Functions. The Chairperson's functions are
   a. To act as the chief representative and advocate for the Department.
   b. To promote and engage in superior teaching, service, and research in the Department.
   c. To plan and administer the Department's budget, in consultation with the Dean and with the advice of the Council.
   d. To make recommendations with respect to faculty appointments, tenure, and promotions, as specified in applicable regulations.
   e. To work for recruitment and promotion of qualified faculty and staff and to ensure equity in hiring and personnel decisions.
   f. To administer all programs of the Department.
   g. To solicit active participation of Department members in Department affairs and to communicate policies.
   h. To preside over meetings of the Council and Assemblies, as necessary.
   i. To appoint ad hoc committees as necessary.
   j. To ensure implementation of this Plan of Organization.

The Chairperson has no vote in any Department Assembly or committee.

Article 2. Constituent Assemblies of the Department

1. The Assemblies. The **General Assembly** of the Department of Computer Science shall be composed of the following constituencies: the Faculty Assembly, the Staff Assembly, the Graduate Student Assembly,

and the Undergraduate Student Assembly. Each voting member of a Constituent Assembly is eligible to vote for the Assembly's representatives for the Department Council.

a. The **Faculty Assembly** shall consist of all full-time University employees who hold at least a half-time academic-year appointment in the Department, with a title classified as faculty in the University's *Faculty Handbook*. This includes faculty on sabbatical leave, but not those on leave-without-pay. Visiting Faculty are not included, nor are faculty researchers who do not hold a Ph.D degree and faculty with teaching duties who do not hold a master's degree in computer science or a related field. A member of the faculty who holds an appointment in the Department with an academic title but who does not meet the description above shall have a voice but no vote in the Faculty Assembly.
b. The **Staff Assembly** shall consist of all non-exempt and exempt staff of the University who hold at least a half-time appointment within the Department.
c. The **Graduate Student Assembly** shall consist of all officially registered graduate students of the University who are enrolled in a graduate program of the Department.
d. The **Undergraduate Student Assembly** shall consist of all officially registered undergraduate students of the University who are declared computer science majors.
e. Any person who belongs to more than one constituency above shall be counted with the category of the person's choice.

2. Functions. Each of these Assemblies serves as an electoral body and as the voice of its members within the Department. Its functions are
   a. To approve the Department Plan of Organization, and any of its Amendments, as well as the Bylaws.
   b. To initiate suggestions to the Department Chairperson through representatives on the Department Council.
   c. To act as the electorate in department, college or campus elections in which the members of that Assembly are allowed to vote.
   d. To identify, where applicable, the Assembly's candidates, either by petition or nomination, for any such election as mentioned in the preceding clause.
   e. To act as referendum body for any referendum called by the Department Council. (All such referenda are advisory in nature.)
3. Organization and Meetings of the Constituent Assemblies.
   a. Each Department Assembly shall be entitled to formulate its own plan which shall not contravene this Plan of Organization or its By-Laws. Any such organizational plan shall be filed with the Department Council.
   b. The Department Constituent Assemblies are not required to hold regularly scheduled meetings. However, upon petition of 20% of the membership of any one Assembly, the Department Chairperson shall call a meeting of this particular Assembly and act as chair pro tempore for such a meeting if the Assembly has no chair at that time.

Article 3. Committees of the Department

1. The **Department Council** shall consist of the Department Chairperson who acts as Chairperson of the Council, along with elected and ex-officio members as specified in the By-Laws.

The functions of the Council are:

   a. To consult with and advise the Department Chairperson on matters of concern to the Department and its Constituent Assemblies.

b. To establish standing and ad hoc subcommittees not otherwise specified in this Plan of Organization as needed or desirable for the Department.
c. To review the Plan of Organization and By-Laws, in years ending with a zero or a five, and to initiate proposed changes then or at other times when necessary or desirable.
d. To perform any other duties specified in the By-Laws.

Meetings of the Council shall be called by the Department Chairperson or upon petition of 25% of the elected Council membership.

2. The **Appointment, Promotion, and Tenure Committee** (APT Committee) of the Department shall consist of an elected APT Chairperson and the Faculty Assembly. Subcommittees of the APT Committee for the various activities of the Committee and the qualifications of the APT Chairperson are specified in the By-Laws.

The functions of the APT Committee and appropriate subcommittees are:

a. To perform an annual review of all assistant and associate professors of the Department. A recommendation to the Department Chairperson concerning their possible promotion, tenure, or continuation of contract will be made as appropriate. Procedures for this are specified in the By-Laws.
b. To review any prospective faculty member for the Department and advise the Department Chairperson on the desirability of the appointment.
c. To perform any other duties specified in the By-Laws.

3. The **Education Committee** of the Department considers all educational matters of the Department. Its membership is specified in the By-Laws. The functions of the Education Committee are

a. To review and recommend changes in the educational programs of the Department.
b. To recommend changes in courses, curricula, admissions and retention of students, and financial aid policy.
c. To establish subcommittees of the Committee as may be necessary or desirable for the performance of its functions.

4. Members of the Faculty Assembly who are tenured, tenure-track, or permanent shall elect **Department Representatives to the University Senate** from their ranks.

5. A **Salary Committee** consisting of three members of the Department tenure-track faculty may be used by the Department Chairperson to assist in determining faculty merit salary raises. Members of this Committee shall be elected by the Department tenure-track faculty.

6. **Other standing or ad hoc committees** may be constituted by the Council, APT Committee, Education Committee, or Department Chairperson as needed or desired to carry out their respective duties, or as required by the University. These committees may be either elected or appointed as deemed appropriate.

Article 4. Elections

All elections for Department Councilors, Committee members, Senators, and APT Chairperson shall be by secret ballot. The By-Laws shall specify the nomination and election procedures, the dates of the elections,

the terms of office of the elected members, and the procedures for filling vacancies in any elected office.

Article 5. Parliamentary Authority

Robert's Rules of Order Revised (most recent edition) shall govern the procedures followed by any assembly, council, committee, or other recognized unit of the Department in all cases in which these rules are applicable. In case of any conflict or inconsistency of the Plan of Organization and By-Laws with duly established and announced rules of the College, Campus, or University System, such rules shall govern.

Absentee ballots are not allowed for any votes at meetings of Assemblies, Council, or committees, but these bodies may decide to use an electronic balloting procedure rather than voting at a meeting.

Article 6. Approval of By-Laws and Amendments to By-Laws

The Department shall have the power to make by-laws and regulations that are in accordance with rules and powers of the College, Campus, or University System. An amendment to the By-Laws may be proposed by any member of the Department. Amendments to the By-Laws shall require approval first by the Department Council and then by at least three of the Constituent Assemblies. Approval by the Council requires a simple majority vote of the entire voting membership of the Council. Approval by any Constituent Assembly shall be by secret ballot and requires a three-fifths majority of those voting on the amendment in that Assembly. The text of the proposed by-law shall be publicly available to each member of the Department at least two weeks prior to the end of the balloting period. Amendments that are approved by this process take effect immediately, unless approval is required by other University bodies.

Article 7. Approval of the Plan of Organization and Amendments to the Plan

The Plan of Organization shall be periodically reviewed by the Department Council as specified above. Amendments to this Plan shall require approval first by the Council and then by each of the Constituent Assemblies. Approval by the Council requires a simple majority vote of the entire voting membership of the Council. Approval by any Constituent Assembly shall be by secret ballot and requires a three-fifths majority of those voting on the amendment in that Assembly. The text of the proposed amendment shall be publicly available to each member of the Department at least two weeks prior to the end of the balloting period. Amendments that are approved by this process take effect immediately, unless approval is required by other University bodies.

By-Laws

Department of Computer Science
University of Maryland
College Park, Maryland

Revised February, 2001

These By-Laws are adopted in accordance with Article 6 of the September, 2000 Plan of Organization of the Computer Science Department.

By-Law 1. Concerning Elections

The Department Council shall be constituted as the Elections Board for all elections. The Elections Board shall specify procedures for the conduct of elections to include and be consistent with the following rules:

1. There will be a single election each year to fill all positions on the Council, all elected positions on other Department Committees, positions for University Senators that need to be filled within the next year, and the APT Chairperson position.

2. Throughout the month before elections, nominations will be sought for undergraduate and graduate student representatives on Department committees. Nominations can be from students or from the Undergraduate or Graduate Coordinator.

3. At the same time, all faculty and staff will be notified that they have been nominated for each position for which they are eligible. Each person nominated will be given the opportunity to decline nomination.

4. Balloting will be by secret ballot, written or electronic. Ballots shall contain only the names of nominated eligible candidates. If there is no nominated eligible candidate for a position, the Department Chairperson will appoint an eligible person to fill the vacancy.

5. Election voting will take place during a two week period and be completed by May 14. The approval voting scheme will be used. Each voter casts one vote for each candidate they find acceptable for the position being voted on. Within a week of completion of voting, the Elections Board will meet to tabulate the vote and announce the winners. If there are k vacant positions within a single category (e.g., Department Council), then the k candidates with the most votes are the winners. (A majority is not necessary.) Should there be a tie vote, the Department Chair will cast the deciding vote.

6. In the case of a tie, the Elections Board shall decide which candidate shall be reported as the winner of the election.

All elected terms of office will commence July 1, unless otherwise specified by University rules.

By-Law 2. Concerning Referenda

The Council may call for a referendum by appointing a subcommittee consisting of at least one representative from each Assembly to conduct the referendum. The subcommittee shall make a copy of the referendum available to all members of the Assemblies during a voting period of at least 14 days and report the result to the Council at the next Council meeting, within a week of the end of the voting period.

By-Law 3. Concerning the Department Council

Faculty, staff, and student members of the Department Council shall be elected by and from the Constituent Assembly they are to represent. In addition to the Council Chairperson, there shall be 7 faculty members, 1 staff member, 2 graduate students, and 1 undergraduate student as councilors. The term of office of each councilor is one year. Councilors may succeed themselves, but may not serve more than two consecutive terms. The Department's Director of Administration shall be an ex-officio non-voting member of the Council and shall act as Council secretary.

If a councilor vacates office before the date of termination of the term, the Department Chairperson shall choose a person from the same constituency as the vacating councilor to fill the remainder of the term.

By-Law 4. Concerning the Appointment, Promotions, and Tenure Committee

All voting members of the Faculty Assembly are voting members of the APT Committee. All other Faculty Assembly members shall be nonvoting members of the APT Committee.

1. Subcommittees

   For the purposes of considering and voting on appointments, promotions and tenure, the following subcommittees are established:

   a. The subcommittee of all Assistant Professors, Associate Professors and Professors considers appointment to the rank of Assistant Professor.
   b. The subcommittee of all Associate Professors and Professors considers contract renewal for Assistant Professors, promotion of Assistant Professors to the rank of Associate Professor, appointment of Associate Professors, and granting the title of Associate Professor Emeritus to an Associate Professor upon retirement. For an appointment or promotion to Associate Professor with tenure, the subcommittee consists of all tenured Associate Professors and Professors.
   c. The subcommittee of all Professors considers promotion of Associate Professors to Professor, appointment of Professors, and granting the title of Professor Emeritus to a Professor upon retirement.
   d. Each of these subcommittees shall be augmented to include research or instructional Faculty Assembly members of equivalent ranks for appointment and promotion considerations of research or instructional faculty members.

2. APT Chairperson

   a. The elected APT Chairperson shall be a Professor (not Assistant or Associate Professor) who holds tenure in the Department.
   b. The APT Chairperson shall be elected by the tenure-track faculty members of the Department, and serve as chairperson for the full APT Committee as well as each of its subcommittees.
   c. The APT Chairperson shall convene the Committee, or respective subcommittee, whenever necessary.
   d. The APT Chairperson shall act as Secretary for each meeting, or shall appoint a Committee or Subcommittee member to act as Secretary. The actions of the Committee or Subcommittee will be in the form of a written report signed by the APT Chairperson including a recommendation to the Department Chairperson expressing approval or disapproval of a proposed action along with the count of the vote. The report to the Department Chairperson shall include any minority reports requested by a committee or subcommittee member.

3. Appointment Considerations

   a. Appointments to the various faculty ranks should meet at least the minimum qualifications as set forth by the University and University Senate.
   b. Such appointments should be based on the individual's teaching, research and service credentials and

the individual's standing in the computer science community.

c. Consideration of faculty with joint appointments with another department are to be coordinated by the Computer Science Department Chairperson with the other department, so that complete information is available to all committees involved.

By-Law 5. Concerning the Education Committee

This committee consists of all members of the Faculty Assembly who have had teaching responsibilities within the last two years, two students elected from the Undergraduate Student Assembly, two students elected from the Graduate Student Assembly, the Undergraduate Student Coordinator, and the Graduate Student Coordinator. The Department Chairperson shall serve as Chairperson of the Education Committee and the Director of Administration shall be an ex-officio member of the Education Committee and serve as Secretary. The Committee may, by majority vote, add other members if appropriate.

Contents

Contents

Past Reviews

Contents

Report of the External Review Committee of the Department of Computer Science at the University of Maryland

D.Ferrari, S.Gerhart, A.Joshi, O.Widlund, P.Young

May 1986

Report of the External Review Committee of the Department of Computer Science at the University of Maryland

D.Ferrari, S.Gerhart, A.Joshi, O.Widlund, P.Young

1. INTRODUCTION

During the last few years, the Department of Computer Science at the University of Maryland has experienced very impressive growth in regular faculty positions, funding for research, and equipment. This extremely rapid expansion has been on the whole managed very successfully. The Department has strengthened its research and teaching programs in several areas, dramatically increased its potential for experimental research in computer science, and hired a large number of high-quality young faculty members. At the same time, the Department has been able to achieve a much needed reduction in the number of its undergraduate majors, and, with further reductions and an expanding faculty, seems to be heading towards an educational load comparable to the average that now characterizes its peer institutions. The recent establishment of UMIACS offers very promising opportunities for strengthening the research program even further with important interdisciplinary components.

This tumultuous growth has not occurred without taking its toll. The most crucial problem it has created is the tremendous shortage of space the Department is suffering from. A shortage of space is a common disease among computer science departments all over the world. In a large number of universities in this country, space is the number one problem in computer science. What makes Maryland's situation particularly worrisome is that the Department is allocated only half of the space that the average institution of similar caliber (which is typically very short of space with respect to its needs) has at the present time in the United States. A survey of computer science facilities recently completed, but not yet published, by the Computer Science Division of the University of California at Berkeley shows that the average department of computer science among the thirty respondents (which included 22 of the top 30 departments in the country) has about 1,000 assignable square feet per regular faculty member. At Maryland, the amount of space per faculty member is approximately 500 ASF.

For the Department of Computer Science, space is not only extremely short: perhaps even worse, it is fragmented. The two sites where departmental space is located are quite distant from each other, and the fact that the Center for Automation Research (where a number of CS faculty and graduate students have their offices) has recently been moved to a third site far from both of the others makes the problem even worse. Keeping all of Computer Science in a single location should be regarded as an extremely important requirement for the Department's welfare. This is particularly true because Computer Science is still a young and relatively fragile discipline, with still rapidly evolving interconnections among subdisciplines. Thus, geographical distance effectively prevents interactions that are absolutely vital to the development of the discipline. Every effort should be made by the University Administration to provide the Department as soon as possible with enough contiguous space for its activities as well as for those of CfAR and UMIACS, which must be kept very tightly coupled with the departmental ones.

Some of the current CS space, especially that in the North Gymnasium, has very poor quality; noise and insufficient ventilation make the working environment quite unpleasant

there. The University should do now whatever it can to improve the environment, and consider very carefully the power, ventilation, and cooling requirements of computer science research in any building that is built or renovated to house the CS Department and the computer-science-oriented research institutes.

Another general problem in computer science departments is faculty overwork, and the CS Department at the University of Maryland is no exception. The normal teaching load of two courses per semester is excessive, even though buying out of courses using research money is a widespread practice which seems to be encouraged by the Department. In most comparable institutions, the normal teaching load is either two or three courses per year, and limited buyouts are allowed, though perhaps not encouraged. As suggested in Section 2 below, the Department could reduce the normal teaching load with positive effects on the graduate program and perhaps some small improvements in faculty workload. Some additional help, though possibly small as well, could be provided by increasing secretarial and grant administration support, as proposed in Sections 3 and 4; for this purpose, soft money could be used while waiting for essential State-funded staff positions. Because of the shortage of senior people coupled with rapid changes in a young and rapidly evolving discipline, it is normal for computer science faculty members at top institutions to be overworked, and there is no effective recipe to make life substantially less hectic for them: the multiple pressures from the students, the sponsors of research contracts, and one's peers in academia and in industry; the administrative needs of university governance in an area in which senior faculty with (or even without) administrative talents are not in abundant supply, the requirements of professional societies, and so on, cannot be avoided. It is, however, an important duty of the Department, the College, and the University to do whatever possible to mitigate these pressures, while at the same time keeping the faculty fully informed and getting it involved in all the important decisions by which their teaching and research work may be affected.

In spite of these problems, and of the others described in the remainder of this report, the External Review Committee found the morale of the faculty, the staff, and the students to be high. [This is certainly due to the substantially correct perception that the Department is an exciting place with a highly positive derivative and a very good, steadily improving overall reputation.]

Following the structure of the Internal Review Committee Report, the rest of this document is divided into four sections. Section 2 addresses our findings and recommendations about the instructional program, Section 3 discusses the research component of the Department's activities, Section 4 the administrative aspects, and Section 5 the Department's external relations.

2. THE INSTRUCTIONAL PROGRAM

The Department is offering a good quality educational program, at both the graduate and undergraduate levels, but the number and variety of graduate level courses needs considerable expansion. The program appears particularly good if one considers the extremely large undergraduate population.

2.1. Area concentrations

Both undergraduate and graduate programs show generally good balance in courses. Particularly commendable is the coverage of software engineering and human/computer interfaces in addition to more standard computer science courses in operating systems, programming languages, numerical mathematics, and theory of computation. However,

hardware and graphics are less well covered, and the Committee feels that the Department should try to improve the situation, especially in an area as fundamental to computer science as that of hardware and architecture. The trend toward increasing the mathematics content of courses is to be commended, but breadth in computer science must be retained.

2.2. Undergraduate majors

The Department handled the dramatic increase in the number of majors and attempted majors well, but perhaps at the expense of the graduate program. Efforts should be made further to decrease the number of majors until a sufficient number of advanced graduate seminars and courses can be offered. It will then be time to restore the Honors program, and commit resources to it.

2.3. Ph.D. course requirements

A student pursuing an M.S. degree may obtain a better education than a Ph.D. student who takes the minimum number of courses, passes the comprehensive exam early, and then takes no further courses. No advanced courses are required, nor are many offered. There is contention for tuition funds coming out of research grants and for faculty time. The students are suffering in both the breadth and the depth of their graduate education. For example, a software engineering Ph.D. from Maryland may have neither "theory" nor "information processing" backgrounds. The Committee recommends that course requirements be added to the Ph.D. program, and that a series of advanced courses in the various fields be planned and offered regularly. A student's specialized education should not be constrained by contention for support from research grants, but should be encouraged by departmental regulations and university support. If the normal teaching load of regular faculty were reduced, and buyouts were not encouraged as strongly as they now are, the research money that would be saved could probably be used to fund tuition waivers for Ph.D. students, thereby allowing them to register more frequently and enroll in required as well as optional graduate courses.

2.4. Comprehensive exams

The comprehensive examination policies have fluctuated for several years. These fluctuations are normal in graduate departments, even though they create additional stress on the students. However, concerns have been expressed to the Committee about the quality and the administration of the examinations, e.g., about the appropriateness of some questions and the preparation of the proctors.

2.5. Fairness

There is a tremendous difference between the treatment of Teaching Assistants and that of Research Assistants. The former often suffer from lack of advising, of space, and of access to equipment. Off-site preparation of textual material is often needed for part-time employed students who do not have accounts on departmental research machines. T.A.s have also suffered from the large undergraduate enrollment. There are morale problems in those students who lack the patronage of a Research Advisor. Since the Department now has considerable amounts of donated and purchased equipment, the Committee recommends that all graduate students be given accounts on a well-supported system, including access to networks and good quality printers. To remedy the lack of advising, all graduate students should be reviewed by the Department at least yearly. In general, the Committee feels that the advising of graduate students could and should be taken more seriously.

Another aspect of fairness is the one having to do with the choice of a research advisor by graduate students. The Department should make sure that the uneven distribution of research funding among the faculty members will not prevent the better graduate students from working with younger faculty.

2.6. Teaching quality

The Department has many fine teachers and researchers. The students interviewed understood the impact of the undergraduate enrollment and appreciated the potential improvements stemming from the addition of new faculty members. However, there were few testimonials to memorable learning experiences.

3. THE RESEARCH PROGRAM

The total amount of funded research (approximately $3.5 M) is quite substantial. This compares favorably with the research funding of many top departments. If one adds, say, 2/3 of the UMIACS funding (as that is roughly the percentage assigned to CS faculty, at present) plus about 70-80% of that of the Center for Automation Research, then the total funding is clearly very substantial (approximately $7.2 M). On a per faculty member basis, the funding is about $87 K (or $180 K if one includes UMIACS and CfAR). These are very good numbers, particularly given that 10 out of the 40 faculty are new and are at the assistant professor level.

Given this amount of research funding, however, the amount of staff support does not seem adequate. Some of the faculty feel the need for planning staff, who could help in alerting faculty who are not directly connected with some of the existing centers or laboratories about current or future opportunities for research support.

3.1. New positions

It would be advisable for the Department to focus now on certain research areas in order to strengthen them. For example, a senior appointment in AI would significantly help that area. There is already considerable expertise in databases, logic programming, and such applied areas as computer vision and robotics. An appointment in the area of planning or knowledge representation would be highly desirable as it would nicely complement the existing expertise. In short, the Department needs to find a highly qualified replacement for Chuck Rieger.

In the theory area, a senior appointment would also provide much needed leadership to a group of very good junior faculty. The coverage of sub-areas in theory is somewhat unbalanced, and care should be taken that it does not become more so. The current emphasis is primarily on the theory of computation.[1] In general, greater breadth within theory should be encouraged. While there seems not even to be any course work on formal languages, the Committee feels that there is an even greater need for more emphasis in courses, examinations, and research in analysis of algorithms. Faculty in the theory area could also benefit from interaction with faculty in parallel processing, vision, and robotics

1. That at the University of Maryland must be one of the few major computer science departments which have a traditional course in mathematical logic (cross listed with Mathematics) taken by a large number of its students. If this is to remain viable, it will be necessary to stabilize the contents of the course and to emphasize topics which will be of interest to faculty and students in such areas as AI and programming languages, as well as theory.

by getting involved in such fields as computational geometry. Unfortunately, the distance of UMIACS and CfAR from the Department does not encourage these interactions.

3.2. Software

In the software area it is also necessary to increase breadth. The faculty should be encouraged to acquire existing software packages, learn to evaluate them and use them to improve their research environment, just as one would use a good compiler. The Department should encourage systems building activities. External distribution of software and even of hardware designs should be regarded as equivalent to publication, and evaluated on the basis of outside letters as well as internal reviews. As in all computer science departments, systems building should not be discouraged among assistant professors because of the fear that they might not publish enough papers to earn tenure.

3.3. Conference papers vs. journal papers

In all fields of computer science, some of the major conferences play a key role as publication outlets. These conferences are highly regarded in the community. Their rejection rates are higher than those of many top journals. Thus, the papers accepted by these conferences should be considered as roughly equivalent to those of the best journal papers in the same field. It is important to keep this in mind in considering promotions of computer scientists. This special role played by selected conferences is found in all areas of CS, but especially in the systems area. The Department seems to understand this special role of publication in the major conferences in computer science, and should continue to keep it in mind in considering promotion and tenure.

3.4. Computer science as an experimental science

Computer science is as much an experimental science as any other among the natural sciences, such as physics or chemistry, or among the engineering disciplines, such as electrical engineering. This means that computer science must have laboratories both for instruction and for research. The CER grant has played a key role in enhancing experimental work in academic computer science. It is extremely important to maintain these laboratories after the CER grant terminates. The commitments (approximately $300K/year) made by the University Administration for these laboratories are substantial, but they are critical and must be honored if the Department is to maintain its infrastructure for experimental research and instruction. Without maintaining this infrastructure, it is doubtful that the Department could maintain its current momentum and continue increasing its overall funding and its national visibility.

4. ADMINISTRATION

The Internal Review Committee of the Department asked the External Review Committee to address four specific questions regarding administrative matters. These were: 1. The level of state support. 2. The number of faculty administrators. 3. How well current space is utilized, and what priorities should be set for the use of existing and new space. 4. Whether part or all of the Department should move off campus to relieve existing pressures. In addition, the External Review Committee considered the role of the departmental field committees, the general departmental organization, and suggestions made by staff.

4.1. Staff

In the discussion with staff members, the group format of the meeting made it difficult to get a comprehensive overview of the full range of staff opinions. Nevertheless some themes emerged. It was, for example, clear that staff members feel overworked and feel that the level of staffing is inadequate. As discussed in Section 3, the Committee also believes that *overall* staffing levels are inadequate given the extraordinarily high student load and the very high volume of research funding in the Department. However, data on individual overloading of various administrative staff members was very hard to evaluate. Nevertheless there was a clear staff consensus that many staff members work overtime (weekends) without compensating pay, and do so on a regular basis. This should be a source of concern.

Staff morale seemed high, but several items of concern require attention. As mentioned in Section 1, the general inadequacy of space compels staff members to work in poorly cooled or poorly heated space, a problem which is often caused by heat generated by workstations and sometimes by inadequate control of machine cooling. The Committee itself found that lack of appropriate air conditioning made their own work room, which often contained workstations, uncomfortable. Furthermore, it appears that inadequate office cooling has resulted in frequent breakdowns of recently donated Xerox workstations and printers. Extreme noise in the areas under the North Gymnasium, and the difficulty of lab staff moving from building to building to service a large number of different systems were also cited by staff members as problems, and legitimately so. Finally, problems with parking and ticketing were a surprisingly strong source of staff complaints.

The undergraduate advising office seems to have done an incredibly good job in dealing with extraordinarily large numbers of undergraduates, and the staff of that office deserves special citation for outstanding service. The graduate advising office appears to be unnecessarily burdened by "walk-in" prospective students who wish to be advised about requirements for admission to the graduate program and to be given an opportunity to plead for special consideration. The Committee believes that this is a common problem in good departments located in urban settings. One solution that has worked well at the University of Washington is to hold meetings once every month or two for prospective graduate students to explain admissions requirements and procedures and to discuss general prospects of students who might apply for admission to the graduate program. At Washington, prospective students from the Seattle urban area are not permitted to talk with advisors at any other time. This has significantly reduced the preadvising load on the graduate office, and prospective students have generally been pleased with the procedure. Given the large number of urban professionals and nonprofessionals who are interested in graduate programs in computer science, a more personal and informal approach to advising prospective students does not seem possible without a significant commitment to advising students who are largely ineligible for admission.

4.2. Departmental administration

Generally, the External Review Committee was impressed by the departmental organization, and by the use of Associate Chairs to distribute the administrative load among faculty administrators. As far as we could see, this system seems to be working well and seems to be based on an appropriate number of departmental officers. The system seems to us to reflect a generally hierarchical approach to departmental administration and operations. Our general sense is that the departmental style is for "top-down" administration, rather than for a perhaps more democratic and decentralized organization. A few individuals expressed some preference for a system, or style, which would bring more faculty into more substantive administrative roles. Although we found little, if any real dissatisfaction with the current style, the Department might want to discuss this issue.

The Committee does want to express concern about the role of the faculty groups within the Department. While some of these groupings, e.g., theory and numerical analysis, may reflect reasonable clusters, the Committee feels that other groupings may reflect archaic partitions of faculty research and educational interests. In some cases these groupings seem to prevent faculty from teaching in areas quite close to their own research interests. In general, the Committee feels that many of the current groups seem, from an external standpoint, unreasonable even for organizing comprehensive examinations, and that such formal groupings often tend to factionalize a department by setting up too rigid compartments. We have no real evidence that this has occurred at Maryland, but we do believe that the current subdivisions are not particularly natural in terms of research and educational objectives, and that in general such groupings tend to work against the need for students and faculty to look for interactions and interconnections between different research areas. We recommend that the Department reconsider the role of faculty groups in general, and the labels and affiliations of some of them in particular.

4.3. University support

The Computer Science Department runs undergraduate and graduate programs which have some of the largest enrollments in the country. Its level of research support easily places it among the top ten departments in the country in terms of total external funding. Given these facts, the current level of university support seems quite inadequate. The adequacy of space has been discussed in Section 1, and will be commented on again below. The number of faculty members is inadequate for the numbers of students served, state-funded secretarial support for faculty members seems almost nonexistent, and apparently only one laboratory staff member is supported from state funds.

When the Department moves into the restructured College of Computing, Mathematical, and Physical Sciences, it will be the dominant academic unit in that College in terms of students served, and it will surely have a significant portion of the total extramural research funding. It should therefore have a number of faculty lines, an amount of faculty and laboratory space, and an allocation of support staff which reflect the size of its research funding and the importance of its educational role within the College.

4.4. Space

The External Review Committee was asked to evaluate the Department's use of existing space, and its priorities for use of this space as well as of the space to be acquired for the future.

We were tremendously impressed by what is being accomplished with the existing space, and with how well this space is allocated and used. And we were happy to learn that the University has some plans for alleviating the Department's desperate need for additional space, which is made even more desperate by the the large number of new faculty members the Department and UMIACS are planning to hire.

The existing arrangement, which puts some of the departmental laboratories, including the Center for Automation Research and the UMIACS offices, in separate, often distant, buildings is largely unacceptable. Computer Science is still a young and relatively fragile discipline, with still rapidly evolving interconnections among subdisciplines. Thus, departmental separation and fragmentation is a severe disadvantage in a computer science department, and particularly in one which is expanding as rapidly as that at the University of Maryland. Furthermore, the increasingly experimental nature of the field requires a heavy commitment to laboratories, particularly for major, multi-investigator grants.

We recommend that the Department and *all* of the associated research centers and institutes be housed in contiguous space. This space should be planned in a way which is adequate for an emerging experimental discipline. Adequate laboratory space is essential, as is adequate space and cooling and electrical outlets for offices which will house a variety of workstations and other types of equipment.

5. EXTERNAL RELATIONS

The Committee has the impression that external relations present certain problems, and that these problems have been somewhat neglected by the Department, at least until recently. This is not surprising in view of the very severe pressures faced by the Department in areas such as overenrollment, space problems, etc. The Department has also been very busy with attracting and managing a rapidly growing volume of sponsored research funding and the building and staffing of a substantial departmental computing facility and other laboratories, which have been made possible through CER funding from the National Science Foundation. The signs of such growing pains are far from unique to this Department.

5.1. UMIACS

The funding of this center represents a real opportunity for the Computer Science Department and the Campus as a whole, but there are also potential problems. Under its acting director, UMIACS appears to have gotten off to a good start. Potentially, the institute could provide much needed stable core funding for research hardware and software support, build an important parallel computing research facility, provide impetus for important cross disciplinary research, and so on. It appears crucial that the offices of the institute remain in close proximity to those of the Department and that the Computer Science faculty play an active role in the search for a Director and in the future advisory structure. There are relatively common misunderstandings in the scientific community as a whole about the nature of computer science research, which is often confused with scientific computing, super computing, and so on. An emphasis on research in core areas of computer science rather than on these other areas seems to be in accordance with the original legislative intent, and the advice of the computer scientists must therefore be heard clearly.

There also appear to be certain other potential problems for the Department. Without close cooperation, UMIACS might build a second group of computer scientists, separated from the Department. It could also factionalize the Department by its ability to decrease the teaching loads and dispense other privileges to a selected subset of the faculty. We emphasize that these are potential, and not existing problems. To date, the impact of UMIACS on the Department has been entirely positive.

5.2. The EE Department

The relations between Computer Science and Electrical Engineering appear to be strained, and additional difficulties could result from the presence of the Systems Research Center. Unfortunately an appointment with a representative of the EE Department could not be arranged for the External Review Committee. The difficulties between the two departments appear to stem in part from the overenrollment in the computer science undergraduate program, which has made it difficult or even impossible for CS to admit EE undergraduates to its courses. A rivalry has naturally developed when the engineers have introduced some parallel courses without much coordination with the Computer Science

curriculum. The relations between the two departments appear to have improved recently. The Committee recommends that the departments explore the possibility of joint educational programs such as a computer science-computer engineering Master's degree. The strengths of the two departments should be complementary, and all should be able to benefit from such an effort. As Computer Science and Electrical Engineering move to different colleges, the University might wish to consider establishing a standing committee to monitor relations between the two departments. Such a committee might also monitor relations between Computer Science and expanding "information science" activities within the Library School.

5.3. Other departments

The Committee was not made aware of any major problems with the Business School. Conflicts between business schools and computer science departments seems to be rarer these days than in the past, and the situation at the University of Maryland does not differ from the one that seems to prevail now.

The numerical analysts in the Department have a fruitful cooperation with the numerical analysts and applied mathematicians in the Mathematics Department and the Institute for Fluid Dynamics. This tripartite arrangement seems to have worked well for the numerical analysts, and has contributed to Maryland's considerable success in this area. There has been good cooperation between the mathematicians and the recently expanded theory group, and undoubtedly there is potential for even more interaction, but it is much too early to tell to what extent this will happen. We were not made aware of any major conflicts with the traditional science departments.

The question of the new college structure of the Campus, and the affiliation of the Computer Science Department with the College of Computing, Mathematical, and Physical Sciences was briefly discussed. The Committee feels that the creation of a separate College of Computer and Information Sciences should be seriously considered.

5.4. The Computing Center

The relations with the Computer Science Center[2] seem to be somewhat uneasy. Not only is the name of this unit a source of confusion, but its choice of hardware and software does not serve the admittedly somewhat special needs of computer science education and research particularly well. A number of software packages important to the computer scientists have not been made available or cannot be made available on the Center's equipment. Editors and other packages available on this equipment are often archaic. Under existing policies the academic departments are expected to pay half of the cost of all but the most standard software. This policy, for which the Center cannot necessarily be blamed, should be changed, or funding should be made available to the Computer Science Department to cover the expenses for its courses. Unless explicitly funded, it is unreasonable to expect the Department to pay the costs of the software required for teaching standard computer science courses.

The lack of graphics equipment should be noted. Many computer science departments are finding that, as the experimental nature of their discipline becomes more apparent, they increasingly need to run a number of specialized instructional laboratories. To date, the Department has only one such laboratory. In the opinion of the Committee, in the near

2. The University of Maryland must be one of the few, if not the only one, in the country which have a computing center with the term "Computer Science" in its name. We strongly recommend a change of name for the Center.

future the Department will need seriously to consider the option of establishing a number of its own instructional laboratories.

5.5. University College

The educational programs in computer science at the University College are believed by some faculty in the Department to be quite poor. Certainly the existence of such programs is likely to be a source of confusion, and their quality could thus reflect poorly on the reputation of the Computer Science Department, which has nothing to do with that activity. We recommend that an appropriate group review the quality of the University College programs.

5.6. Industry

The relations with industry through the industrial affiliates program and the Advisory Board seem to be excellent and contribute importantly to the Department in both political support and much needed unrestricted funds.

Evaluation of the Computer Science Department and the Institute for Advanced Computer Studies at the University of Maryland

Evaluation Committee:
Mani Chandy, Gene Golub, Susan Graham, Ed Lazowska, Shmuel Winograd

The committee met at the University of Maryland from March 1 through March 3, 1993. After an initial meeting with the Dean, the first day was devoted to UMIACS: we met with the director, most of the permanent faculty, and several groups of rotating faculty. The second day was devoted to CSD: we met with the chair, the council, many of the faculty (in a half dozen group meetings), undergraduate and graduate students, and the chairs of some related departments. The third day was devoted to organizing our report and to an exit meeting with the Dean. The committee is grateful for the hospitality provided by the college, UMIACS, and CSD.

Our report is organized around the following sections:

1. Structure: This section identifies structural strengths and weaknesses of CSD and UMIACS. By structural issues we mean issues dealing with reporting structures, control of budgets, and control of resources such as space.

2. Culture: Each institution develops a *culture* over time. The culture of an institution is a collection of behavior patterns of its members. In this section we identify those aspects of the culture of CSD and UMIACS that help and do not help in the mission of the college and the university.

3. Individuals: The success of an institution depends on individuals, though structure and culture also play a role. This section is a brief discussion of the role of some of the dedicated individuals playing leadership roles in CSD and UMIACS.

4. Recommendations: We make suggestions for improving the structure and culture to better fit the overall mission of the university.

To put this report in perspective, we state at the outset that computer science activity at the University of Maryland is very strong – the department is highly ranked, and many faculty are internationally recognized as leaders in their research areas. We believe, however, that the value of this report lies mainly in the identification of areas where changes can make a very good activity even better. For that reason the report discusses weaknesses and remedies out of proportion to their place in the overall picture.

Structure

The discussion about structure is divided into separate subsections for UMIACS and CSD; for each subsection, we first consider the strengths and then the weaknesses.

UMIACS

Strengths

1. Interdisciplinary: UMIACS has been successful in its efforts to foster interdisciplinary research. There are several success stories demonstrating genuine collaborative research between computer scientists and people from Linguistics, Business, Government and EE. The people we talked to, who were participating in interdisciplinary research, said that the research would have been difficult to carry through without UMIACS. In many cases the research would not have been initiated without UMIACS. Clearly UMIACS is pursuing its mission of fostering interdisciplinary computing research at College Park.

2. Common Space: Though many people feel that the most exciting research in computing is at the interstices between fields, getting people to collaborate across fields is difficult. The approach taken by UMIACS of housing collaborators in the same building is an excellent one. The university is very lucky to have space that can be used in this manner.

3. Leverage: UMIACS plays an important role at getting research groups together to leverage resources to achieve common goals such as getting large parallel computers (e.g., the CM5) at the university. Here too, UMIACS is succeeding in its role of being the catalyst for active collaboration both within the university and even reaching out to institutions outside the university.

4. Terrific Staff: We heard unanimous praise for the staff at UMIACS. UMIACS provides valued support in preparing grant proposals, administering grant accounts, organizing conferences, and many other important activities. UMIACS staff has been invaluable in providing services that CSD staff cannot provide because CSD staff are too few and have too many other obligations.

5. Outreach: UMIACS has helped the university to reach out to the larger community of scholars, and to government and industry. UMIACS has helped to organize

several important and well-attended conferences. UMIACS participates in the activities of some U.S. government agencies and it has the opportunity to do even more.

6. **Support for CS Departmental Growth and Retention:** Without UMIACS budgetary support, the department could not have grown rapidly in the mid-80s. UMIACS support has also been very helpful in retaining people in CSD during the last two or three difficult years.

7. **Support for EE Department Growth and Retention:** UMIACS has helped the EE department by supporting some of its faculty doing computer-related research.

8. **Successful Recruiting:** UMIACS played an important role in recruiting people to CSD and EE.

Weaknesses

1. **Conflicting Missions:** UMIACS has multiple conflicting missions, including the following:

   (a) Help CSD to grow by allocating UMIACS funds to CSD faculty so that CSD can hire more faculty.

   (b) Reward the faculty doing the best research in CSD and EE.

   (c) Serve as a catalyst for increased interdisciplinary computing research at the College Park campus.

   People in different positions in the administration have differing opinions about the relative priorities of the conflicting missions. Also, these priorities seem to have changed over time: when UMIACS started, the priority of helping CSD grow seemed paramount; later, the priority of rewarding faculty doing the best research became the most important; now, the interdisciplinary mission is stressed. The lack of a mission statement, clearly articulated to the entire College Park campus, is a severe problem.

   During times of growing budgets, conflicting missions can be satisfied to some extent. Now, at times of stagnant or even decreasing budgets, a clear mission is essential.

2. **UMIACS Controls CSD Rewards:** During the recent years of extremely tight budgets, CSD has had very limited resources to reward faculty doing exceptional work or starting new courses, new research directions, or community outreach. UMIACS has worked to reward CSD faculty. The structural problem is that the reward system for one organization is controlled by another. CSD has missions, such as undergraduate teaching and minority recruiting, that are not central to UMIACS. UMIACS is expected to use its resources to further its own mission. CSD needs to control its own resources and to use them in a flexible way to achieve its own goals.

3. **Two-Class Citizenry:** Some faculty are supported 75% by UMIACS, some 50% and some less, some are supported on a permanent basis, some on a temporary basis, and some not at all. This creates a feeling of a class structure, in addition to (and orthogonal to) the traditional professorial hierarchy.

   Most faculty understand the process for promotion in the traditional hierarchy. This process is usually spelled out in writing, and involves deliberations at all levels of the administration. The criteria employed by UMIACS to appoint faculty to permanent positions and to rotating positions, and to determine the degree of support (25%, 50%, 75%), are documented and understood far less well, subject to far less review, and potentially far more subjective (for instance, the expected impact of proposed research as evaluated by a small group).

   Many faculty seem to feel that the criteria for a position in UMIACS overlap with the criteria for promotion in the traditional university hierarchy: excellence in research and a strong reputation in one's research area. The presence of two hierarchies, both concerned with research excellence, is confusing. This problem will decrease if a unique mission for UMIACS, different from that of CSD, is articulated, and if it is possible to reward research excellence outside of UMIACS.

4. **Limited Flexibility:** The Dean and the higher administration seem to be placing emphasis on UMIACS' mission to engender interdisciplinary research in computing. To meet this objective, UMIACS needs to be flexible to seize new opportunities that arise, to drop projects that become self-supporting, and to also drop projects that are unsuccessful. UMIACS' flexibility is limited by the funds it must provide to its permanent faculty, and the funds it must provide to CSD by way of rotating faculty to keep CSD afloat. Dynamic response to a rapidly changing environment requires more flexibility.

5. **Inadequate Intellectual Leadership in Interdisciplinary Research:** If a major role of UMIACS is to spearhead interdisciplinary research in computing across the College Park campus, then the management of UMIACS must play the major role in identifying thrust areas and in bringing about collaborative research. The Director has played an energetic and valuable role in this effort. It is far less clear, though, that the remaining management of UMIACS (including permanent members and long-term rotating members) has been active in identifying new areas of opportunity and aggressively bringing about collaborations. We feel that UMIACS faculty should take the initiative in identifying new exciting interdisciplinary areas (whether or not they themselves will work in those areas) and should then either stimulate activity in these areas or drop them after due deliberation. *UMIACS faculty should be viewed by faculty across the College Park campus as being the visionaries of interdisciplinary computing research.*

   The governance structure of UMIACS is at variance with the emphasis on spearheading interdisciplinary computing research. UMIACS does not require its faculty to provide intellectual leadership in interdisciplinary computing. The faculty are rewarded for continuing their ongoing research, and doing it well. In particular, permanent faculty positions are offered to people doing excellent research in their own areas. The strategic plan for UMIACS seems very close to the union of the individual agendas of the permanent members. As such, it appears to be intended to justify their own UMIACS funding.

   In the present structure, the permanent members and long-term rotating members also administer the orthogonal mission of engendering interdisciplinary computing research. Given the increasing emphasis on the mission of fostering interdisciplinary computing research, there must be more attempts to identify new directions for the College Park campus.

6. **The Primary Reward is Teaching Buyout:** The primary incentive for participation in UMIACS seems to be reduced teaching. That situation is sometimes at odds with the needs of the department. A consequence of the focus on rewarding the best research with reduced teaching is that some of the department's stronger research areas have an insufficient number of courses.

   Given the interdisciplinary mission of UMIACS, we would have expected the primary reward to be more funds focused on incubating new interdisciplinary efforts.

CSD

Strengths

1. Undergraduate Education: The undergraduate education program is impressive. Most core courses are taught by professors – not instructors, visitors, or TAs. Some faculty do a tremendous job of being available, inviting undergraduates to participate in their research groups, giving advice regarding courses and careers, and generally being friendly. A small number of faculty names were mentioned by several of the undergraduates as being truly helpful. This small core of CSD faculty has done a terrific job in improving undergraduate education to the point that the entire campus can be proud of its CS undergraduate education. Many Maryland undergraduates go on to graduate education in the very best schools, and compete for top positions in industry.

2. Strong Groups: The department has several strong groups in a variety of areas. Some of the groups are quite large (in addition to being strong) and, in many cases, collaborative.

3. Strong Individuals: The department has very strong individuals in many areas.

4. Effective Council: The council has become more effective in the past several years. In the past, the council was used primarily as a mechanism for delivering announcements. Now, the council is used to determine direction (as, for example, in strategic planning) and to build consensus. Most importantly, the council has become increasingly proactive, identifying opportunities and warning of potential problems, instead of being merely reactive.

5. InterCollege Activity: The department and its chair have been proactive in setting up a computer engineering program with EE despite perceived difficulties in cooperating across two colleges.

6. Support for Young Faculty: The department has been helpful to young faculty in setting up laboratories and similar ventures despite its limited staff and shrinking budgets.

7. Adequate TA Support: The department seems to have adequate (although certainly not extravagant) TA support – but we note that 50% of TA positions are funded from unfilled faculty positions.

Weaknesses

1. **Lack of Consensus and Ineffective Governing Structures:** The department does not appear to have a common vision. As a consequence, building a consensus on any issue is difficult. Examples of issues for which consensus building has either taken many years or has not succeeded include changing the Ph.D. comprehensive exams, and deciding the areas in which to recruit new faculty.

   To a visitor, the structure of CSD seems closer to a loose federation of units for different areas (A.I., Systems, Theory, Software Engineering,...) than an integral whole. The governance of a federation is difficult in the best of times, but is extremely difficult in times of diminishing resources.

   Decision-making in a federation is slower and more conservative than in an integral unit. Building consensus, without a common vision, requires a great deal of time. When decisions are made without total consensus, there is insufficient commitment from all members to the group to uphold the decision against inevitable dissension. As a consequence, strife ensues.

2. **Excessive Dependence on Counting in Evaluations:** We noticed a reluctance to evaluate research from the point of view of long-term impact, and a preference to use "objective measures" such as numbers of papers, numbers of grants, amounts of dollars and so on. Though counting has value as a measure of activity, it is not an ideal measure of impact. A few seminal papers open up new areas of research, and CSD should attempt to nurture people who write such papers.

   The preference for counting may also be a consequence of a lack of a common vision. Subjective measures require that the evaluators agree on what is important.

3. **Slow to Move into New Areas:** One strategy for a good department to improve is to focus on unique synergistic opportunities that derive from the location of the department – its proximity to government, to industry, and to other strong departments on campus. A related strategy is that of developing areas that are likely to become important in the future, and placing less emphasis on mature areas. These strategies are risky, but conservative strategies are not likely to propel a department ranked in the second 10 into the top 10.

The department seems slow to move into new areas. Again, this may be a consequence of a lack of consensus on what exciting areas will emerge, as well as an evaluation structure that provides disincentives for current faculty to change areas.

4. **Dependence on UMIACS:** The department is unable to control its budget because of its dependence on UMIACS financial support. The department has inadequate resources and inadequate flexibility to reward its members who embark on new directions.

5. **Graduate Program Not High Priority:**

   (a) Recruiting to the graduate program seems passive. Recruiting in most schools with which Maryland competes is much more active; departments actively search for students and then do their best to recruit them.

   (b) Graduate students are not adequately mentored after they arrive until they pass their comprehensive exams and join a research group. Incoming students are given advisors, but the students seem to be in limbo until they jump over the much-feared hurdle of comps.

   (c) The graduate core curriculum seems stagnant. This is especially surprising given CSD's active research program.

   (d) The 700-level courses are not well-rounded. There are many research seminar courses, and there are the standard core courses, but not enough in between.

   (e) The comprehensive exams are viewed as a barrier with little educational value. Comprehensive exams are feared in most departments, but graduating Ph.D.s often admit that the exams had a significant educational role. In CSD, the educational role of the comprehensives, as perceived by most Ph.D. students, seems very small; they seem to think of the comprehensives almost exclusively as a filter.

   (f) It is clear that the production of top quality Ph.D. students (students recruited by the very best academic departments and industrial research labs) does not play a major role in the department's evaluation and reward system. (We shall have more to say about the department's evaluation and reward system, which we feel requires serious attention, in a later section.)

6. **Joint Appointments:** The absence of interdepartmental joint appointments with EE and other departments is a lost opportunity.

7. **Course Load:** The official course load of 4 courses over 2 semesters is above the national average of 2.9 courses per year. This forces the department to use accounting tricks with UMIACS funding to reduce the course load. A cleaner structure would be to have a course load comparable to the national average without the need to rely on UMIACS.

8. **Minorities:** We did not find evidence of aggressive proactivity in recruitment and retention of minority undergraduate and graduate students. The D.C. area should present unique opportunities in this regard.

9. **Relationship with CfAR:** The vision group and CfAR are among the strengths of the College Park campus. The "arms length" relationship between CfAR and CSD is a lost opportunity for synergy.

10. **Salary Compression:** Three or more years of no salary raises, or very low raises, causes several problems. A serious problem is that of salary inversion and compression. The faculty should feel that rewards are fair, and some faculty members in CSD feel that rewards have not been fair.

    There are many issues in determining salaries of incoming faculty, including paying a salary attractive enough to enable the candidate to participate in the relatively expensive suburban D.C. housing market. We are not criticizing salary decisions made by CSD. The structural problem is that several years of small raises, coupled with aggressive recruiting, hurts morale if all the faculty do not subscribe wholeheartedly to these decisions. For the most part, CSD faculty understood the problems faced by the university and the department, and appreciated the actions taken by the college and the department in an extremely difficult situation. There were, however, some exceptions.

Culture

The discussion about culture embraces both UMIACS and CSD. We first consider the strengths and then the weaknesses.

Strengths

1. High Activity: Almost everyone is very active in one or more of research, education, and administration. The department seems to have a tradition of activity among its members.

2. Collaboration: There is significant collaborative activity within groups inside CSD. There is relatively little across groups (A.I., Systems, Theory, Software Engineering...) though that is changing, and certainly UMIACS had done a great deal to further the change.

   We do not advocate collaborative activity as an end in itself, but we are concerned about cultural barriers to collaboration.

3. Outreach: There are new and valuable outreach programs to the community such as the K-12 program.

4. Individual Initiative: Individual initiative is allowed to flower. Faculty members have started work in new areas, new collaborative efforts have been started in recent years, some faculty members have put in a lot of energy into the undergraduate program.

5. Strong Junior Hires: The faculty recruits, in recent years, are strong. The recruiting program seems to have become much more thoughtful, proactive, and selective than in earlier years of extremely rapid growth.

6. Departmental Organization: The overall departmental organization (for instance, a council with student members and members from all ranks) seems inclusive and good. The organization can be used effectively given more goodwill and unity of vision.

7. Growing Interdisciplinary Work: We noticed increasing attempts at interdisciplinary work in several areas. The computational science educational projects of Stewart and O'Leary are examples.

Weaknesses

1. **Parochialism Bordering on Selfishness:** The department seems similar to a federation of little groups. Faculty members seem to feel strong "ownership" of their groups, but much less ownership to the department, college, or campus as a whole. This lack of ownership is strikingly evidenced by the many faculty, including senior ones, who referred to the department as "they" rather than "we."

   The culture does not seem to foster a sense of belonging to the larger group. All members of a group must feel a sense of participation, belonging, ownership and pride for the group to move forward rapidly.

   The lack of community spirit is evident in the difficulty in sharing resources, deciding areas within which to recruit, and the reward system generally.

2. **Desire to Avoid Teaching:** The desire to avoid teaching by almost all of the faculty was striking. As best as we could tell, the distaste for classroom teaching did not seem to be a consequence of inadequate TA support or poor classrooms, but seemed to be part of the culture. The primary reward for excellent performance, beyond advancement through the traditional university professorial hierarchy, seems to be reduced teaching responsibilities. Reductions of teaching loads to the national average are certainly understandable, but there seems to be strong desire among large numbers of faculty to teach even less than one course each semester.

3. **Excessive Conservatism:** The reluctance to change is remarkable. Conservatism has its place, but institutions in areas as dynamic as computing must adapt to new areas and to new kinds of research ventures. The lack of change to the core curriculum is a case in point. The excessive conservatism is due in part to structural problems, and in part to a feeling of complacency.

4. **Demoralization:** There is evidence of some demoralization. There is also evidence of disunity. The egregious breach of electronic mail security following our visit to campus is a case in point.

   Some are reluctant to contribute to the general good. A few are active and vocal about feeling left out.

   These problems are partly due to the restricted levels of resources, and partly to UMIACS' control of the CSD reward structure, but there are deeper causes as well – to a disturbing extent, disunity has become institutionalized.

Individuals

We were not asked to evaluate individual members of the faculty or staff. We wish to emphasize, however, that CSD and UMIACS have very strong researchers who are doing excellent work. An evaluation of institutions is incomplete without some mention of its leaders. The heads of UMIACS and CSD seem to us to be doing excellent jobs in difficult circumstances. Next, we discuss our perceptions of the leadership of Larry Davis and Satish Tripathi based on conversations with heads of other similar units, faculty, and students.

1. Larry Davis is doing an excellent job of helping to start interdisciplinary research activities. Several faculty members referred to Larry's personal efforts as being instrumental in starting and maintaining interdisciplinary computing research. Larry played an important role in getting the CM5 to Maryland by putting together a consortium of groups and organizations; such collaborative efforts will become increasingly important. Allocating space for interdisciplinary research in a single building, and encouraging meetings and workshops, is also something that Larry has pushed for. Many faculty members said that Larry was a person who responded to requests with "Let's see how we can make this happen". A positive personality, who encourages new directions and new ways of doing things, is a definite plus for the UMIACS director.

   The structural and cultural problems faced by a director of UMIACS have been outlined already. A case in point is that Larry exercises relatively little direct control on who gets appointments to UMIACS or the level of support. A committee puts applicants into three categories, and Larry exercises his judgment mostly for the (small) middle category. Larry plays the role of catalyst more than director.

   Larry has limited flexibility in managing UMIACS: the permanent members' budget is fixed, a certain amount of money has to be given to CSD to keep CSD afloat, and the appointments to UMIACS are determined by committee (for the most part). Despite the limited flexibility and multiple conflicting missions, Larry has done an excellent job of leading UMIACS.

2. Satish Tripathi has been doing an excellent job in a very difficult situation. Among the very worst possible situations for a chair is the situation in Maryland: a period of rapid and somewhat uncontrolled growth with unbounded optimism and the hiring of a large number of faculty of variable quality, followed by a period of flat budgets and tenure-denials, and the continued recruiting of faculty

at (perforce) competitive salaries with ensuing salary compression problems. This coupled with the federated structure of the department and its dependence on UMIACS makes the position of chair extremely difficult.

Satish has been trying to build a consensus for the major decisions made by the department. It is not possible, or desirable, to involve the entire department in every decision such as negotiating salary for a potential recruit. Satish appears to have attempted to be inclusive in decision-making. Total consensus is probably impractical in a department as large and diverse as CSD. Yet, decisions without total consensus seem to create strife and there is insufficient support from the rest of the faculty in reducing tensions. Satish seems to have done everything he can to help create a harmonious climate. The rejuvenated council is a case in point.

Some faculty members said that they would like Satish be more authoritative, and spend less time identifying consensus. A more authoritative head can make decisions faster and move into new areas. An authoritative head, however, needs *active* support from the large majority of the faculty. There is evidence that many faculty members, including some senior members, are not actively engaged in the governance of the department. In this situation, it is difficult for a chair to act authoritatively. Satish seems to have done an excellent job, in difficult circumstances, perhaps spending more time building consensus than is necessary at other institutions.

3. Other Individuals: CSD and UMIACS are blessed with many other strong individuals, some of whom actively participate in the life of these institutions (in addition to teaching their courses and doing good research). We decided not to identify these people because our identification may not be complete.

We wish to emphasize that a department with a large number of strong individuals, each of whom does his or her "own thing" well but is not otherwise actively engaged in the life of the institution, is not likely to move rapidly into the ranks of the top departments in the country.

Recommendations

UMIACS

1. Mission: Define a clear and succinct mission for UMIACS. This mission should be distinct from the mission of the department. Since the mission of the department is excellence in teaching and research in computer science, the mission of UMIACS must be something else. A possible mission is to serve as a catalyst for interdisciplinary research in computing across the College Park campus.

2. Budget: Provide adequate resources to meet the mission. Split the existing budget into two parts, one managed by CSD and the remainder managed by UMIACS. The part managed by CSD is to meet CSD's mission of excellence in computer science teaching and research. This part also must include the funds used by CSD to hire faculty. The part managed by UMIACS is for the sole purpose of meeting UMIACS' more focused research mission, including the fostering of interdisciplinary research in computing.

   (Our charge was to review UMIACS and CSD. We recognize that EE, too, has a special relationship with UMIACS, although not on the scale of CSD's. We have not addressed that relationship in this report.)

3. Strategic Plan: Have UMIACS produce a strategic plan to meet its more focused mission. In particular, the plan should identify areas of opportunity across the entire campus, paying particular attention to the strengths of other departments and to the unique opportunities provided by the D.C. area. The plan should be much more than a collection of the individual plans of the permanent members of UMIACS. The plan should have visibility at the level of the Provost and President.

4. Advisory Board: Activate the advisory board for UMIACS. The advisory board should help the director meet the mission of interdisciplinary research across the College Park campus.

5. Permanent Appointments: Do not limit the flexibility of the "new UMIACS" by making further permanent appointments or otherwise committing funds on a permanent basis. To the extent that permanent memberships are deemed to

play the role of chaired professorships rather than fulfilling the UMIACS mission, consider decoupling the two roles and reconstituting UMIACS membership.

6. **Clarify Selection Procedure:** Clarify the procedure by which appointments are made to UMIACS, and funds are provided by UMIACS. The evaluation criteria can be subjective, but the criteria must be formulated and disseminated. The criteria must not be identical to the criteria used by the department in promotion, because dual-reward procedures for identical criteria are confusing. Presumably, the criterion will be that of furthering UMIACS' mission.

7. **Provide Intellectual Leadership in Interdisciplinary Research:** The director and long-term members of UMIACS must be seen by people all across the College Park campus as providing the intellectual leadership in interdisciplinary computing. This requires the long-term members to have a much broader interest in computing than continuing to do their excellent research. The long-term members must engage other people on campus in active dialog with a view to furthering computing research. This activity currently is being carried out by the director, but the campus must also be able to look to the long-term members for this leadership.

8. **Reduce Emphasis on Teaching Buyout:** The primary reward of UMIACS should be the opportunity and facilities to participate in research, particularly in interdisciplinary research. It is the responsibility of CSD and the other departments to determine teaching loads. The director may decide that a temporary reduced teaching load is necessary for a particular project, and may request that reduction from the individual's department, but this should be a decision for a specific project and not a broad-based consequence of UMIACS membership. Benefits of UMIACS other than teaching load include office space in an interdisciplinary community, excellent shared equipment and staff, visibility across campus and the D.C. area, funding for graduate students and postdocs, and access to students and colleagues with knowledge in computer science and other areas. Indeed, rather than distancing itself from teaching, UMIACS can help start new interdisciplinary courses.

9. **Resources:** Make no further cuts in UMIACS. If cuts must be made, the mission must be scaled down.

CSD

1. Graduate Program: The department must realize that its success rests in large part on its graduate program. *All members of the department must help in improving the graduate program.* Some members can be given the specific task of revamping the graduate program, and people with this task should be given time off from teaching because this task is directly related to the teaching mission.

   The quality of graduate students should be improved by better financial support, aggressive recruiting, and refusing to admit students who seem less than excellent. In the short run, limiting graduate enrollment may cause hardships, but in the long run decreased variability in the quality of graduate students will help the department in many ways.

2. Teaching: The culture must be changed to view teaching as a rewarding experience. Course loads should not exceed the national average, but teaching should be regarded as an important and valued part of the job. The department must find out why so many of the faculty avoid teaching, and then take steps to turn that situation around.

   We feel that as a matter of general policy, all faculty (including permanent members of UMIACS) should teach two courses per year, and that all faculty should regularly teach both undergraduate and graduate courses.

3. Reduced Emphasis on Counts in Evaluations: The department and UMIACS should reduce the emphasis on counting – counting papers, counting dollars, counting numbers of grants, etc. – in evaluating faculty performance. What matters is *impact.* Impact can be assessed, but it cannot be counted. The department and UMIACS should be willing to use subjective criteria such as likely impact on the broad area of computing over a significant period of time. An emphasis on counting can have a number of deleterious side-effects, including faculty who work in increasingly narrow areas and are unwilling to try new research directions.

4. Nurture Interdisciplinary Work: The mission of CSD is excellence in teaching and research in computing, and not specifically interdisciplinary activities. On the other hand, the department can benefit from the synergy provided by UMIACS by

also nurturing interdisciplinary work. In our view, a key value of interdisciplinary work is that it facilitates the identification of "high leverage" computer science research problems – problems whose solutions really matter to someone. It should be every researcher's goal to work on high leverage problems.

5. **Permanent Faculty:** The permanent UMIACS faculty should be encouraged to become much more actively involved in the general welfare of both UMIACS and their departments. The permanent UMIACS faculty were appointed for their excellence in research. They also have an obligation to further interdisciplinary research. The permanent members should be the leaders in CSD and across campus in teaching and research.

6. **The Computer Science Center:** The name of the Computer Science Center should be changed. It's very confusing.

7. **Resources:** The funding level and associated resources for CSD should be sustained. Additional funds should be provided, as soon as possible, to help alleviate stresses in CSD.

8. **Strategic Plan:** The strategic plan worked out by the department is a good first step. The department should attempt to chart out a course for itself over the next five years, even though the future is uncertain. Everybody in the department should feel ownership in the plan. Then, the department should change the plan on a regular (perhaps yearly) basis, and measure its progress in terms of meeting its plan.

The committee wishes to re-emphasize that UMIACS and CSD are very strong organizations doing very good work. This report is not balanced – it deals primarily with problem areas and potential remedies. The committee feels privileged to have worked with groups as strong as UMIACS and CSD.

Contents

Department Research Accounts
FY 98 - 02

AGENCY	PI	DURATION	AWARD AMOUNT
NSF	Arbaugh	9/1/02 - 8/31/07	$ 488,188
DOD - ARPA	Arbaugh	6/26/01 - 6/25/03	$ 585,113
Universities Carnegie Mellon	Basili	12/15/01 - 12/14/02	$ 385,000
NSF	Basili	9/15/00 - 8/31/03	$ 2,108,455
NSF	Basili	9/15/00 - 8/31/03	$ 7,000
NASA - Goddard	Basili	2/15/00 - 11/14/02	$ 538,541
NSF	Bhattacharjee/Keleher	9/15/01 - 8/31/04	$ 709,974
NSF	Bhattacharjee	8/15/98 - 7/31/03	$ 200,154
NSF	Bhattacharjee	8/12/01 - 8/31/06	$ 500,000
NSF	Chawathe	8/25/00 - 9/30/03	$ 253,026
NSF	Chawathe	8/25/00 - 8/31/03	$ 451,528
Corporations Fraunhofer	Davis	1/1/98 - 6/30/03	$ 168,168
NSF	Davis	7/1/01 - 6/30/03	$ 73,959
NSF	Elman	7/15/99 - 6/30/03	$ 130,000
NSF	Elman	8/1/02 - 7/31/03	$ 122,000
DOD - Navy.ONR	Elman/O'Leary	12/1/00 - 11/30/03	$ 201,088
Universities Dartmouth	Hendler	3/1/02 - 2/28/03	$ 66,305
Universities Tennessee	Hollingsworth	9/15/01 - 9/14/03	$ 303,739
Universities UC Lawrence Livermore Labs	Hollingsworth	5/7/99 - 4/30/03	$ 175,000
Universities Wisconsin at Madison	Hollingsworth	8/1/96 - 7/31/02	$ 970,719
NSF	Hollingsworth	3/1/97 - 8/31/02	$ 242,800
DOE - Chicago	Hollingsworth	9/15/01 - 9/14/03	$ 501,553
NSF	Keleher	9/1/00 - 8/31/03	$ 111,988
NSF	Keleher	9/1/02 - 8/31/03	$ 150,000
NSF	Khuller	6/1/99 - 7/31/03	$ 270,111
NSF	Khuller	8/1/01 - 7/31/03	$ 6,500
NSF	Khuller/Golubchik	8/1/01 - 7/31/03	$ 242,504
NSF	Mount	9/1/01 - 8/31/03	$ 170,002
DOD - Air Force	Nau	1/30/00 - 4/25/04	$ 625,440
DOD - Navy	Nau	4/8/02 - 4/7/04	$ 90,000
Universities	Porter	12/31/99 - 8/31/03	$ 199,337
NSF	Porter	9/1/01 - 8/31/03	$ 182,312
NSF	Porter	9/1/02 - 8/31/03	$ 107,871
Universities Purdue	Pugh	5/30/01 - 5/29/05	$ 468,105
NSF	Pugh	9/1/01 - 8/31/04	$ 307,624
DOD - Navy	Purtilo	3/1/99 - 6/30/05	$ 224,000
HHS - PHS/NIH DIVRESGRANT	Reggia	9/1/96 - 7/31/05	$ 1,305,223
NSF	Saltz	3/1/00 - 2/28/03	$ 426,272
NSF	Samet	8/15/99 - 7/31/03	$ 66,000
NSF	Smith	9/1/01 - 8/31/04	$ 209,999
NSF	Smith	6/13/02 - 8/31/04	$ 6,000
NSF	Srinivasan	6/1/02 - 5/31/05	$ 203,357
NSF	Stewart/O'Leary	6/15/02 - 6/30/05	$ 510,000

DOD - Army	Subrahmanian	8/1/00 - 1/31/03	$	265,105
Universities Ohio State	Sussman	6/1/01 - 5/31/03	$	43,855
NSF	Sussman	10/1/01 - 9/30/03	$	105,000
NASA - Goddard	Sussman	5/1/02 - 4/30/03	$	100,000
NSF	Tseng	9/1/01 - 8/31/04	$	215,000
Universities George Mason	Tseng/Pugh	4/1/02 - 3/31/03	$	345,000
NSF	Varshney	9/1/00 - 8/31/03	$	450,000
NSF	Varshney	8/22/00 - 6/30/03	$	183,890
			$	16,772,805

10/31/2002

Contents

Contents

UMIACS Research Accounts

AGENCY	PI	DURATION	AWARD AMOUNT
Bowie State/NSF	JaJa/Davis	9/15/99-8/31/02	284,558
Chevron	Plaisant/Bederson	11/15/00-8/31/03	271,000
JHU/ARL	Resnik/Weinberg/Dorr	5/1/01-4/20/04	1,062,184
UCSD/ARL	JaJa	10/1/99-9/30/03	335,369
UCSD/NSF	JaJa	10/1/97-9/30/03	2,715,800
Univ New Mex/ARL	Chellappa/VS	6/10/00-9/30/09	1,721,000
General Dyn	Davis/Chellappa/Aloimonos	6/10/00-9/30/09	3,300,000
ITT/Kaman	Basili	12/15/97-12/11/02	198,343
Phillips	Davis	6/1/98-5/31/01	72,454
Mitre	Dorr/Weinberg	11/20/01-1/30/03	189,000
Foster-Miller	Davis	7/1/01-2/28/03	130,000
Keck	Yorike-Davis	12/1/00-6/30/03	100,000
BBNT	Dorr	3/1/02-2/28/03	50,000
GDRS	Davis	3/14/02-5/30/02	196,000
Honda	Davis	1/1/02-12/31/02	32,000
Raytheon	Davis/Chellappa/Weiss	8/10/02-9/30/03	125,000
Precept Tek	Davis	9/15/02-7/15/04	59,100
Samsung	Arbaugh	6/1/02-5/31/02	49,443
IMLS	Druin/Bederson/Weeks	10/1/02-9/30/05	397,162
Univ Balto/NSF	Bederson/Druin	7/1/02-12/31/03	133,125
MERL	Davis	9/1/02-6/30/03	15,000
HHS	Shneiderman/Plaisant	8/1/02-5/31/03	40,000
NSF	Basili/Zelk	8/15/97-7/31/01	902,256
NSF	Samet	3/15/99-2/28/02	166,278
NSF	Basili/Zelk	8/1/99-7/31/02	79,706
NSF	Samet	9/1/99-8/31/02	500,000
NSF	Davis/Chellappa/Hollingsworth/ Kelleher/Aloimonos/Sussman	9/15/99-8/31/02	1,096,011
NSF	Davis/Duraswami	4/1/00-3/31/02	700,000
NSF	Yeung/Tseng	7/1/00-6/30/02	319,999
NSF	Davis/Duraswami/Chellappa/ Elman/Shamma/Aloimonos	9/1/00-8/31/04	2,999,995
NSF	Aloimonos	10/1/00-9/30/03	199,999
NSF	Keleher/Hollingsworth/Pugh/ Roussopoulos/Saltz	9/15/00-8/31/03	861,244
NSF	Samet	9/1/00-8/31/03	519,999
NSF	Samet/Golubchik/Khuller	6/15/01-5/31/04	616,172
NSF	Raschid/Doermann/Oard/Dorr	9/15/01-8/31/04	132,006
NSF	Plaisant/Shneiderman	6/1/02-5/30/03	182,366
NSF	Duraiswami/Davis/Shamma	9/1/02-8/31/03	1,800,000
NSF	Druin/Bederson/Weeks	9/1/02-10/30/07	3,000,000
NSF	Gumerov/Duraswami/Elman/Oleary	10/1/02-9/30/02	293,249
NASA	JaJa/Townshend	3/1/98-6/30/03	5,036,966
NIST	Arbaugh/Agrawala	9/30/01-9/29/05	861,236
AF	Davis/VS	9/20/99-9/19/02	500,000
AFOSR	Perlis	12/1/99-11/30/02	494,301
AFOSR	Perlis	7/1/99-12/31/02	299,331
ARL	VS/Hendler/Druin	5/19/97-3/19/02	3,200,000
ONR	Dorr/Hendler/Oard/Resnik	6/13/97-6/30/01	1,077,204
ARO	VS	9/16/02-9/15/03	20,000
Navy/SPAWAR	Oard/Dorr/Resnik/Doermann	1/3/00-12/30/03	2,370,000
ONR	Chellappa/Davis/Aloimonos	8/1/00-12/31/02	1,922,393
ONR	Yacoob/Davis	12/1/99-11/30/02	288,760
ONR	Yacoob/Davis	12/1/02-11/30/05	351,184
ONR	Reggia	7/1/02-6/30/03	200,000
NIJ	Davis/Chellappa	9/1/99-6/30/01	353,870
NSA	Davis/Doermann	8/25/00-8/24/02	798,380
NSA	JaJa	1/14/99-1/14/02	2,799,286
NSA	JaJa	5/24/01-6/30/04	4,923,431
Census	Norman/Plaisant/Shneiderman	9/1/00-6/30/01	300,000
		TOTAL	**51,642,160**

Contents

Private Sector Funding

The members of our Department have many relationships with corporations. Interactions include support of individual faculty member's research programs, gifts to support graduate fellowships, and general donations to the Department.. In addition, many faculty members have active collaborations with companies that include sending students for summer internships and hosting extended visits by employees from companies.

In terms of financial contributions to the department (and to UMIACS in support of CS faculty), over $2,217,000 has been donated since January 1, 2000. Of that total, $1.9 million was in support of faculty research, $120,000 was in the form of fellowships for students, and $65,000 was in support of department wide activities (including the department lecture series and high school outreach programs). The amount given by various companies is shown in the table below. Funds to support individual faculty members were given to ten faculty (Agrawala, Arbaugh, Bederson, Davis, Door, Hendler, Hollingsworth, Pugh, Shneiderman, and Shankar). The total amount of corporate cash support per faculty member ranged from $25,000 to $1,012,000.

Company	**Amount**
Fujitsu	1,000,000
Chevron	216,000
Microsoft	195,000
IBM	132,000
Aerospace	70,000
AOL	70,000
Sun Microsystems	70,000
BBNT	50,000
Hughes Network	50,000
Lockheed Martin	50,000
Ricoh	50,000
Toshiba	50,000
Honda	32,000
Panasonic	27,000
SAIC	25,000
Intel	5,000
Koolspan	5,000
Total	**2,217,706**

In addition to cash contributions, significant in-kind donations of hardware and software have been received. The total value of donated hardware since January 2000 is over $500,000. Examples of in-kind donations include an IBM Power 4 system (valued at $400,000), several multiprocessor Ultra-Sparc III systems, and an ultra-high resolution monitor from IBM. In addition to outright donations of hardware, many vendors sell equipment to the department at deep discounts (frequently 60-90% off list price).

Contents

Faculty teaching fall 1998 through spring 2003

Below are the names of courses taught by each professor or instructor/lecturer over the past ten semesters. The number of courses listed for each faculty member is less than ten primarily because most of our faculty taught the same course for more than one semester. In some cases the number is less because some of our faculty accepted positions at other institutions, some became faculty here during this time, and others took sabbaticals or did buy-outs.

Professorial faculty

Ashok Agrawala
- CMSC411 Computer Systems Architecture
- CMSC412 Operating Systems
- CMSC417 Computer Networks
- CMSC818Z Information-centric Design of Systems

Yiannis Aloimonos
- CMSC426 Image Processing
- CMSC733 Computer Processing of Pictorial Information
- CMSC828Z 3D Photography and Inverse Rendering

William Arbaugh
- CMSC414 Computer and Network Security
- CMSC818A Advanced Computer and Network Security
- CMSC818Y Computer Security

Victor Basili
- CMSC435 Software Engineering
- CMSC735 A Quantitative Approach to Software Management and Engineering

Ben Bederson
- CMSC427/828E Computer Graphics
- CMSC434 Introduction to Human-Computer Interaction
- CMSC838B Zoomable Interfaces
- CMSC498B Developing User Interfaces

Samrat Bhattacharjee
- CMSC417 Computer Networks
- CMSC412 Operating Systems
- CMSC711 Computer Networks

Sudarshan Chawathe
- CMSC424 Database Design
- CMSC724 Database Management Systems

CMSC828C Webology/XML

Larry Davis
CMSC420 Data Structures
CMSC426 Image Processing
CMSC818D Fundamentals of Computer Vision

Bonnie Dorr
CMSC150(250) Discrete Structures

CMSC297 Honors Seminar
CMSC421 Introduction to Artificial Intelligence
CMSC828D Issues in Large Scale Natural Language Processing Applications

Howard Elman
CMSC460 Computational Methods
CMSC661 Scientific Computing
CMSC666 Numerical Analysis I
CMSC667 Numerical Analysis II
CMSC760 Advanced Linear Numerical Analysis
CMSC878E Mathematics of Finite Element Methods

Christos Faloutsos
CMSC424 Database Design

Manoj Franklin (affiliate)
CMSC411 Computer Systems Architecture

Michael Franklin
CMSC424 Database Design
CMSC624 Database Systems Implementation
CMSC828F Reading in Database Systems

John Gannon
CMSC330 Organization of Programming Languages

William Gasarch
CMSC150(250) Discrete Structures
CMSC451 Design and Analysis of Algorithms
CMSC456 Cryptology
CMSC752 Concrete Complexity

Richard Gerber
CMSC417 Computer Networks

Lise Getoor
CMSC828G Principles of Data Mining

CMSC421 Introduction to Artificial Intelligence

Leana Golubchik
- CMSC417 Computer Networks
- CMSC420 Data Structures
- CMSC710 Performance Evaluation of Computer Systems

Francois Guimbretiere
- CMSC434/828S Introduction to Human-Computer Interaction
- CMSC838G Advanced Topics in HCI

James Hendler
- CMSC422 Robotics
- CMSC498X Xtreme Programming: The Semantic Web
- CMSC828Y Artificial Intelligence on the Web
- CMSC421 Introduction to Artificial Intelligence

Michael Hicks
- CMSC433 Programming Language Technologies and Paradigms
- CMSC838Y Agile and Adaptive Programming Systems

Jeffrey Hollingsworth
- CMSC412 Operating Systems
- CMSC417 Computer Networks
- CMSC714 High Performance Computing
- CMSC818Z Advanced Topics in Computer Systems

Liviu Iftode
- CMSC412 Operating Systems
- CMSC818I Advanced Topics in Computer Systems
- CMSC818L Network-centric Systems

David Jacobs
- CMSC426 Image Processing

Jonathan Katz
- CMSC456 Cryptology
- CMSC858K Cryptography

Peter Keleher
- CMSC411 Computer Systems Architecture
- CMSC412 Operating Systems
- CMSC414 Computer Security
- CMSC818K Peer-to-Peer Computing

Samir Khuller
- CMSC251(351) Algorithms

CMSC451 Design and Analysis of Algorithms
CMSC651 Analysis of Algorithms
CMSC858K Theory of Computing

Clyde Kruskal
CMSC150(250) Discrete Structures
CMSC251(351) Algorithms
CMSC311 Computer Organization
CMSC411 Computer Systems Architecture
CMSC456 Cryptology
CMSC751 Parallel Algorithms
CMSC858G Advanced Topics in Theory of Computing
CMSC858K Theory of Computing

Atif Memon
CMSC435 Software Engineering
CMSC838M Advanced Topics in Software Testing

Raymond Miller
CMSC452 Elementary Theory of Computing
CMSC818M Advanced Topics in Systems

David Mount
CMSC420 Data Structures
CMSC427 Computer Graphics
CMSC451 Design and Analysis of Algorithms
CMSC754 Computational Geometry
CMSC828K Data Structures and Algorithms for Information Retrieval

Dana Nau
CMSC420 Data Structures
CMSC421 Introduction to Artificial Intelligence
CMSC722 Artificial Intelligence Planning

Dianne O'Leary
CMSC411 Computer Systems Architecture
CMSC460 Computational Methods
CMSC660 Scientific Computing I
CMSC666 Numerical Analysis I
CMSC764 Advanced Numerical Optimization

Donald Perlis
CMSC251(351) Algorithms

Adam Porter
CMSC433 Programming Language Technologies and Paradigms
CMSC435 Software Engineering

CMSC838P/Q Object-Oriented Design Patterns for Distributed and Concurrent Systems

William Pugh
- CMSC433 Programming Language Technologies and Paradigm
- CMSC631 Program Analysis and Understanding
- CMSC731 Programming Lanaguage Implementation
- CMSC838P Advanced Topics in Programming Lanaguages

James Purtilo
- CMSC435 Software Engineering
- CMSC498C Special Problems In Computer Science
- CMSC498P Special Problems In Computer Science

Louiqa Raschid (affiliate)
- CMSC828R DBChat: Readings in Database Management Systems

Philip Resnik (affiliate)
- CMSC723 Natural Language Processing

James Reggia
- CMSC421 Introduction to Artificial Intelligence
- CMSC726 Machine Learning
- CMSC727 Neural Modeling
- CMSC498R/828R Evolution Computation and Artificial Life

Azriel Rosenfeld (affiliate)
- CMSC828C Computer Vision Seminar

Nick Roussopoulos
- CMSC424 Database Design
- CMSC624 Database System Implementation
- CMSC724 Database Management Systems
- CMSC828R Database Research in the 21st Century

Joel Saltz
- CMSC424 Database Design
- CMSC615 Advanced Computer Architecture

Hanan Samet
- CMSC420 Data Structures
- CMSC725 Geographical Information Systems and Spatial Databases

Udaya Shankar
- CMSC412 Operating Systems
- CMSC417 Computer Networks
- CMSC712 Distributed Algorithms and Vertification

Ben Shneiderman
- CMSC434/828S Human Factors in Computer and Information Systems
- CMSC838S Seminar in Human-Computer Interaction

Carl Smith
- CMSC150(250) Discrete Structures
- CMSC251 Algorithms
- CMSC452 Elementary Theory of Computing
- CMSC650 Theory of Computing

Aravind Srinivasan
- CMSC451 Design and Analysis of Algorithms
- CMSC858S(CMSC858T) Randomized Algorithms

Gilbert (Pete) Stewart
- CMSC411 Computer Systems Architecture
- CMSC417 Computer Networks
- CMSC466 Introduction to Numerical Analysis I
- CMSC662 Computer Organization and Programming for Scientific Computing
- CMSC667 Numerical Analysis II
- CMSC760 Advanced Linear Numerical Analysis

V. S. Subrahmanian
- CMSC420 Data Structures
- CMSC424 Database Design
- CMSC498S/CMSC828T(CMSC828V) Software Agents
- CMSC720 Logic for Problem Solving

Alan Sussman
- CMSC330 Organization of Programming Languages
- CMSC411 Computer Systems Architecture
- CMSC433 Programming Language Technologies and Paradigms
- CMSC818S Parallel and Distributed Data Intensive Computing

Chau-Wen Tseng
- CMSC430 Theory of Language Translation
- CMSC732 Compiling for High Performance Architectures
- CMSC838T High Performance Computing and Bioinformatic Applications

Amitabh Varshney
- CMSC427 Computer Graphics
- CMSC740(CMSC828V) Advanced Computer Graphics
- CMSC828W Seminar in Computer Graphics

Uzi Vishkin (affiliate)
- CMSC751 Parallel Algorithms

Amy Weinberg (affiliate)
- CMSC723 Natural Language Processing

Marvin Zelkowitz
- CMSC330 Organization of Programming Languages
- CMSC430 Theory of Language Translation
- CMSC630 Theory of Programming Languages

Instructional faculty

John Arras
- CMSC102 Introduction to Information Technology
- CMSC250 Discrete Structures

Fawzi Emad
- CMSC106 Introduction to C Programming
- CMSC114 Computer Science I
- CMSC250 Discrete Structures

James Glenn
- CMSC114 Computer Science I
- CMSC150(250) Discrete Structures
- CMSC214 Computer Science II

Evan Golub
- CMSC102 Introduction to Information Technology
- CMSC214 Computer Science II
- CMSC434 Introduction to Human-Computer Interaction

Laurence Herman
- CMSC106 Introduction to C Programming
- CMSC114 Computer Science I
- CMSC214 Computer Science II
- CMSC330 Organization of Programming Languages

Michelle Hugue
- CMSC311 Computer Organization
- CMSC411 Computer Systems Architecture
- CMSC420 Data Structures

Charles Lin
- CMSC106 Introduction to C Programming
- CMSC214 Computer Science II
- CMSC250 Discrete Structures
- CMSC298L Effective Programming
- CMSC311 Computer Organization

Sangeeta Maheshwari
- CMSC114 Computer Science I

James Maybury
- CMSC106 Introduction to C Programming
- CMSC114 Computer Science I
- CMSC214 Computer Science II
- CMSC298X UNIX

Nelson Padua-Perez
- CMSC114 Computer Science I
- CMSC214 Computer Science II
- CMSC298P Introduction to Graphical Programming

Jandelyn Plane
- CMSC102 Introduction to Information Technology
- CMSC106 Introduction to C Programming
- CMSC107 Introduction to the UNIX Operating System
- CMSC250 Discrete Structures

Brian Postow
- CMSC106 Introduction to C Programming
- CMSC150(250) Discrete Structures
- CMSC214 Computer Science II
- CMSC351 Algorithms

Steve Scolnik
- CMSC106 Introduction to C Programming

Bunny Tjaden
- CMSC214 Computer Science II

Part-time and postdoc faculty

James Beisaw
- CMSC421 Introduction to Artificial Intelligence

Samrat Chandran
- CMSC420 Data Structures

William Cheng
- CMSC420 Data Structures

Leila Defloriani
- CMSC828L Geometric and Solid Modeling
- CMSC420 Data Structures

J. Fekete
CMSC838F Information Visualization

Robert Fourney
CMSC414 Computer and Network Security

Mark Glezer
CMSC114 Computer Science I

Scott Henninger
Knowledge Management for Software Development

Howard Killam
CMSC434 Human Factors in Computer and Information Systems

Jeremy Manson
CMSC106 Introduction to C Programming

Hector Munoz-Avila
CMSC420 Data Structures

Sam Noh
CMSC411 Computer Systems Architecture

David Preston
CMSC311 Computer Organization
CMSC330 Organization of Programming Lanaguages

Richard Salter
CMSC434 Human Factors in Computer and Information Systems

Douglas Szajda
CMSC411 Computer Systems Architecture
CMSC417 Computer Networks

R. Thurimella
CMSC420 Data Structures

G. Travassos
CMSC435 Software Engineering

V. Zadorozhny
CMSC424 Database Design

Contents

Faculty Honors

Fellows, American Association for the Advancement of Science (AAAS)

Laveen Kanal
Raymond Miller
Jack Minker
Ben Shneidermann

Fellows, American Association for Artificial Intelligence (AAAI)

James Hendler
Jack Minker
Laveen Kanal
Dana Nau
Azriel Rosenfeld

Fellow, American College of Medical Informatics

James Reggia

Fellows, Association for Computing Machinery (ACM)

Victor Basili
Ben Shneiderman
Raymond Miller
Hanan Samet
Jack Minker
Uzi Vishkin
Azriel Rosenfeld

Fellows, International Association of Pattern Recognition (IAPR)

Larry Davis
Hannan Samet
Laveen Kanal
Azriel Rosenfeld

Fellows, Institute of Electrical and Electronic Engineers (IEEE)

Ashok K. Agrawala
Raymond Miller
Victor R. Basili
Jack Minker

Ramalingam Chellappa
Azriel Rosenfeld
Larry Davis
Hanan Samet
Joseph Jaja
Marvin Zelkowitz
Laveen Kanal

National Science Foundation Presidential Faculty Fellow Award

Bonnie Dorr

National Science Foundation Career Awards

Bill Arbaugh
Samrat (Bobby) Bhattacharjee
Sudarshan Chawathe
Christos Faloutsos (now at CMU)
Michael Franklin (now at UC Berkeley)
Leana Golubchik
Jeffrey Hollingsworth
Pete Keleher
Samir Khuller
Adam Porter
Dieter Rombach (now at Fraunhoffer Institute, Germany)
Timos Sellis (Not at Technical University, Athens)
Chau-Wen Tseng
Amitabh Varshney

National Science Foundation Presidential Young Investigator Awards

John Yiannis Aloimonos
Howard Elman
Dana Nau
William Pugh
James Reggia
V. S. Subrahmanian

Packard Fellow

Bill Pugh

Alfred Sloan Fellowship

Bonnie Dorr

Another indication of our Department's national impact is the many exceptional honors and awards received by faculty. A few recent examples are:

Fullbright scholars/Fellowships: Smith, Hendler
IBM Faculty Partnership Awards: Arbaugh, Hollingsworth
ACM SIGCHI Lifetime Achievement Award: Shneiderman
ACM SIGSOFT Outstanding Research Award: Basili
ACM SIGSOFT Distinguished Service Award: Zelkowitz
F.L. Bauer Prize: Stewart

ITR proposals submitted 2003

1. Basili/Zelkowitz/Cukier/Arbaugh/Memon, "Collaborative Research in Empirical Evaluation and Evolution of Validation Techniques to Improve Software Security"
2. Basili/Zelkowitz, "Collaborative Research on Investigating and Evolving the Nature and Structure of the Empirical Research Engine for Software"
3. Getoor, "Schema Discovery, Integration and Reformulation"
4. Keleher/Bhattacharjee/Chawathe/Katz, "Robust, Distributed Indexing and Searching in Untrusted, Peer-To-Peer Environments"
5. Nau/Subrahmanian/Gupta, "A General Framework for Plan Management"
6. Reggia, "Self Organizing Collective Intelligence for Adaptive Problem Solving"
7. Pugh/Hollingsworth/Memon/Hicks/Foster, "Tools to Improve Software Reliability Using a Shared Repository"
8. Roussoupoulos, "Collaborative Research: GaiaNET: A Peer-To-Peer Web Service Composition System for Supporting Wide-Area Scientific Database Applications"
9. Hendler, "Science and the Semantic Web – Prototypes in Biology"
10. Gligor/Baras/Khurana/Arbaugh/Katz, "Security of Emergent Properties In Ad-Hoc Networks"
11. Mount/Gupta, "Assessing Shape Similaritiy of 3D Objects Using Projections of Feature Vectors
12. Reggia/Frels, "Standard-Scape an Agent-Based Model of Competition In Technology Markets With Network Externalities
13. Subrahmanian, "Technology Obsolescence Modeling for Risk-Informed Optimization"
14. Hendler/Getoor, "Advanced Zoological Information Management System"
15. Aloimonos/R. Etienne-Cummings, "Polydiptric and Argus Eyes: New, Biologically Inspired Cameras with New Capabilities"
16. Davis/Sussman, "Human Imaging with Ultra High Field Resonance"
17. Dorr, "Interlingual Annotation of Multilingual Text"
18. Mount/Duraiwaswami/Gumerov, "Fast Multipole Algorithms"
19. Perlis/D. Nelson, "Metacognitive Agents"

20. Samet, “The Urban Eco-Atlas”
21. Vishkin/Barua/Qu/Tseng, “PRAM-on-a-Chip”
22. Yeung/Tseng, “Novel Subordinate Multithreading Techniques and Architectures”

Graduate course offerings:

Algorithms and Computation Theory

CMSC 451 Design and Analysis of Computer Algorithms
CMSC 452 Elementary Theory of Computation
CMSC 456 Data Encryption and Security
CMSC 475 Combinatorics and Graph Theory
CMSC 650 Introduction to Computability(Inactive Course)
CMSC 651 Analysis of Algorithms
CMSC 750 Advanced Theory of Computation(Inactive Course)
CMSC 751 Parallel Algorithms
CMSC 752 Concrete Complexity
CMSC 753 Linear Programming and Combinatorial Optimization
CMSC 754 Computational Geometry

Artificial Intelligence

CMSC 421 Introduction to Artificial Intelligence
CMSC 720 Logic for Problem Solving
CMSC 721 Non-Monotonic Reasoning
CMSC 722 Artificial Intelligence Planning
CMSC 723 Natural Language Processing
CMSC 726 Machine Learning
CMSC 727 Neural Modeling

Database Systems

CMSC 420 Data Structures
CMSC 424 Database Design
CMSC 624 Database Systems Implementation
CMSC 724 Database Management Systems
CMSC 725 Geographic Information Systems and Spatial Databases

Computer Vision and Graphics

CMSC 420 Data Structures
CMSC 426 Image Processing
CMSC 427 Computer Graphics
CMSC 725 Geographic Information Systems and Spatial Databases
CMSC 740 Computer Graphics
CMSC 733 Computer Processing of Pictorial Information
CMSC 754 Computational Geometry

Scientific Computing

CMSC 460 Computational Methods
CMSC 466 Introduction to Numerical Analysis I
CMSC 660 Scientific Computing I
CMSC 661 Scientific Computing II
CMSC 662 Computer Organization and Programming for Scientific Computing
CMSC 666 Numerical Analysis I
CMSC 667 Numerical Analysis II
CMSC 760 Advanced Linear Numerical Analysis
CMSC 762 Numerical Solution of Nonlinear Equations
CMSC 764 Advanced Numerical Optimization

Programming Languages/Software Engineering & Human-Computer Interaction

CMSC 430 Theory of Language Translation
CMSC 433 Programming Language Technologies and Paradigms
CMSC 434 Human Factors in Computer and Information Systems
CMSC 435 Software Engineering
CMSC 630 Formal Methods
CMSC 631 Program Analysis and Understanding
CMSC 632 Software Product Assurance
CMSC 731 Programming Language Implementation
CMSC 732 Compiling for Vector and Parallel Architectures
CMSC 735 Quantitative Approach to Software Management and Engineering
CMSC 736 Software Engineering Environments

Systems and Networks

CMSC 412 Operating Systems
CMSC 414 Computer Security
CMSC 417 Computer Networks
CMSC 710 Performance Evaluation of Computer Systems
CMSC 711 Computer Networks
CMSC 712 Distributed Algorithms and Verification
CMSC 713 Real-time Systems
CMSC 714 High Performance Computing

Our 400-level courses are primarily senior undergraduate courses which can also count as graduate courses. They are offered every semester. The intention is that graduate students should take them only to acquire needed background. The 600-700 level courses are the core, long-standing curriculum. They are offered on a rotating basis, each roughly every three or four semesters. In addition to the courses listed above, special topics courses are offered, under the course numbers CMSC 498, 798, 818, 828, 838, etc. Courses from other departments may be used for MS/Phd qualifying coursework after approval.

Graduate seminar courses

Since computer science is a rapidly changing field, we use 800-level courses as a mechanism for rapidly introducing new graduate courses into our curriculum. Often these courses are offered multiple times, and may subsequently become permanent courses. Such courses are approved by field committees on a case-by-case basis as to whether they count towards the students' required PhD course requirements. Listed below are the 800-level courses we have offered between Fall 2000 and Fall 2002 inclusive:

Algorithms/Theory

858G Parallel Algorithms
858K Approximation Algorithms
858S Randomized Algorithms

Artificial Intelligence

828D Fundamentals of Computer Vision
828D Advanced NLP: Theory and Practice
828G Data Mining (crosslisted in DB)
828R Evolutionary Comp. & Artificial Life
828S Software Agents
828Y Semantic Web

Databases

828G Data Mining (crosslisted in AI)
838L Information Retrieval Systems

Image Processing, Graphics

828C Computer Vision Seminar
828V Advanced Computer Graphics
828D Fundamentals of Computer Vision
828L Geometric and Solid Modeling
828Z 3D Photography and Inverse Rendering

Programming Languages, Software Engineering, Human Computer Interactions

828S Human Factors in Computer and Information Systems
838B Information Visualization
838M Advanced Topics in Software Testing
838S Human Factors in Computer and Information Systems
838V Knowledge Management for Software Development

Systems

818K Advanced Operating Systems
818K Peer-to-Peer Computing
818L Network-Centric Systems
818M Topics on Communication Protocols
818S Parallel and Distributed Data Intensive Computing
818Z Information-centric Design of Systems

Results of the Graduate Survey

A.1. The curriculum is broad enough to meet my needs and prepare me for my chosen career.

Strongly Agree	12	Average: 3.23	Count: 113
Agree	63		
Neutral	25		
Disagree	12		
Strongly Disagree	1		
Don't know or not applicable	1		
No Response	0		

A.2. Courses are offered with sufficient regularity to meet my program needs.

Strongly Agree	7	Average: 0.80	Count: 113
Agree	46		
Neutral	27		
Disagree	24		
Strongly Disagree	9		
Don't know or not applicable	1		
No Response	0		

A.3. The courses I have taken provide me with valuable knowledge for my chosen career and future research.

Strongly Agree	15	Average: 3.86	Count: 114
Agree	72		
Neutral	16		
Disagree	8		
Strongly Disagree	3		
Don't know or not applicable	0		
No Response	0		

A.4. Overall, the workload in courses I have taken has been

Much too high	4	Average: 2.08	Count: 113
Somewhat high	46		
About right	56		
Somewhat low	7		
Much too low	0		
No Opinion	1		
No Response	0		

A.5. Overall, the number of students in the courses I have taken has been

Much too high	8	Average: 1.55	Count: 110
Somewhat high	20		
About right	80		

Somewhat low	2
Much too low	0
No opinion	4
No Response	0

A.6. Overall, the grading in courses I have taken has been

Much too harsh	3	Average: 0.76	Count: 112
Somewhat harsh	23		
About right	75		
Somewhat lenient	10		
Much too lenient	1		
No opinion	2		
No Response	0		

A.7. The department does a good job of preparing graduate students for:
(a) academic careers

Strongly Agree	7	Average: 2.21	Count: 104
Agree	47		
Neutral	36		
Disagree	13		
Strongly Disagree	1		
Don’t know or not applicable	9		
No Response	1		

(b) careers outside of academia

Strongly Agree	3	Average: 0.45	Count: 100
Agree	32		
Neutral	42		
Disagree	17		
Strongly Disagree	6		
Don’t know or not applicable	14		
No Response	0		

B.1. The course requirements for an M.S. degree are reasonable in terms of

(a) the number of courses required

Strongly Agree	11	Average: 4.38	Count: 73
Agree	46		
Neutral	13		
Disagree	2		
Strongly Disagree	1		
Don’t know or not applicable	5		
No Response	36		

(b) the breadth of coverage

Strongly Agree	9	Average: 3.56	Count: 73
Agree	45		
Neutral	10		
Disagree	7		
Strongly Disagree	2		
Don't know or not applicable	4		
No Response	37		

(c) the course grade requirements

Strongly Agree	16	Average: 4.32	Count: 73
Agree	37		
Neutral	15		
Disagree	4		
Strongly Disagree	1		
Don't know or not applicable	3		
No Response	38		

(d) the time required until completion

Strongly Agree	9	Average: 4.31	Count: 72
Agree	48		
Neutral	12		
Disagree	2		
Strongly Disagree	1		
Don't know or not applicable	5		
No Response	37		

B.3. The course requirements for a Ph.D. degree are reasonable in terms of
(a) the number of courses required

Strongly Agree	9	Average: 2.22	Count: 108
Agree	61		
Neutral	14		
Disagree	17		
Strongly Disagree	7		
Don't know or not applicable	1		
No Response	5		

(b) the breadth of coverage

Strongly Agree	11	Average: 2.41	Count: 108
Agree	59		
Neutral	16		

Disagree	15
Strongly Disagree	7
Don't know or not applicable	1
No Response	5

(c) the course grade requirements

Strongly Agree	10	Average: 1.90	Count: 108
Agree	51		
Neutral	25		
Disagree	14		
Strongly Disagree	8		
Don't know or not applicable	1		
No Response	5		

(d) the time required until completion

Strongly Agree	11	Average: 2.04	Count: 108
Agree	50		
Neutral	23		
Disagree	20		
Strongly Disagree	4		
Don't know or not applicable	1		
No Response	5		

C.1. As a teaching assistant, I was adequately trained before entering the classroom.

Strongly Agree	6	Average: -0.19	Count: 77
Agree	16		
Neutral	22		
Disagree	25		
Strongly Disagree	8		
Don't know or not applicable	5		
No Response	32		

C.2. As a teaching assistant, I received appropriate supervision to help improve my teaching skills

Strongly Agree	3	Average: -1.12	Count: 76
Agree	18		
Neutral	23		
Disagree	23		
Strongly Disagree	9		
Don't know or not applicable	5		
No Response	33		

C.3. My professional needs and interests are given appropriate consideration for determining the courses to which I was assigned.

Strongly Agree	8	Average: -0.20	Count: 74
Agree	19		
Neutral	19		
Disagree	18		
Strongly Disagree	10		
Don't know or not applicable	6		
No Response	34		

C.4. My workload as a teaching assistant was reasonable

Strongly Agree	5	Average: 2.05	Count: 78
Agree	39		
Neutral	19		
Disagree	13		
Strongly Disagree	2		
Don't know or not applicable	3		
No Response	33		

C.5. The teaching experience offered through the department is adequate preparation for an academic/teaching career.

Strongly Agree	3	Average: 0.33	Count: 92
Agree	30		
Neutral	34		
Disagree	20		
Strongly Disagree	5		
Don't know or not applicable	5		
No Response	17		

D.1. Graduate students in the department receive adequate quality office and laboratory space

Strongly Agree	14	Average: 0.58	Count: 112
Agree	40		
Neutral	19		
Disagree	23		
Strongly Disagree	16		
Don't know or not applicable	2		
No Response	0		

D.2. Graduate students in the department receive adequate access to computer facilities

Strongly Agree	22	Average: 3.72	Count: 113
Agree	56		
Neutral	22		
Disagree	10		
Strongly Disagree	3		

Don't know or not applicable	0
No Response	1

D.3. Graduate students in the department receive adequate access to library facilities

Strongly Agree	15	Average: 4.09	Count: 110
Agree	68		
Neutral	21		
Disagree	4		
Strongly Disagree	2		
Don't know or not applicable	3		
No Response	1		

D.4. Graduate students in the department receive adequate administrative support

Strongly Agree	24	Average: 4.68	Count: 110
Agree	58		
Neutral	25		
Disagree	3		
Strongly Disagree	0		
Don't know or not applicable	3		
No Response	1		

D.5. Graduate students in the department receive adequate financial support for professional travel

Strongly Agree	2	Average: -0.97	Count: 88
Agree	22		
Neutral	32		
Disagree	21		
Strongly Disagree	11		
Don't know or not applicable	26		
No Response	0		

E.1. Graduate students in the department receive adequate training in professional skills such as:
(a) public speaking

Strongly Agree	5	Average: -1.14	Count: 105
Agree	26		
Neutral	25		
Disagree	38		
Strongly Disagree	11		
Don't know or not applicable	9		
No Response	0		

(b) writing research papers

Strongly Agree	9	Average: 1.02	Count: 108

Agree	42
Neutral	26
Disagree	24
Strongly Disagree	7
Don't know or not applicable	6
No Response	0

(c) writing grant proposals

Strongly Agree	1	Average: -5.11	Count: 89
Agree	5		
Neutral	18		
Disagree	32		
Strongly Disagree	33		
Don't know or not applicable	23		
No Response	2		

E.2. Graduate students in the department receive effective guidance and placement assistance and job search support

Strongly Agree	1	Average: -2.01	Count: 77
Agree	15		
Neutral	26		
Disagree	22		
Strongly Disagree	13		
Don't know or not applicable	37		
No Response	0		

F.1. The department encourages student involvement in research at an early stage.

Strongly Agree	6	Average: -0.50	Count: 111
Agree	38		
Neutral	24		
Disagree	25		
Strongly Disagree	18		
Don't know or not applicable	3		
No Response	0		

F.2. The department provides me with clear, regular assessments of my progress towards my degree.

Strongly Agree	4	Average: -1.58	Count: 111
Agree	21		
Neutral	37		
Disagree	34		
Strongly Disagree	15		
Don't know or not applicable	3		
No Response	0		

F.3. The department provides graduate students sufficient and regular funding to facilitate adequate progress towards their degree.

Strongly Agree	11	Average: 2.19	Count: 105
Agree	46		
Neutral	29		
Disagree	16		
Strongly Disagree	3		
Don't know or not applicable	9		
No Response	0		

G.1. My advisor provides me with ongoing, constructive feedback on my progress.

Strongly Agree	24	Average: 3.14	Count: 105
Agree	42		
Neutral	19		
Disagree	16		
Strongly Disagree	4		
Don't know or not applicable	8		
No Response	1		

G.2. I am satisfied with the time I spend with my advisor in terms of:

(a) the degree of access to my advisor

Strongly Agree	26	Average: 4.20	Count: 106
Agree	45		
Neutral	30		
Disagree	2		
Strongly Disagree	3		
Don't know or not applicable	7		
No Response	1		

(b) the amount of time I spend with my advisor

Strongly Agree	21	Average: 2.88	Count: 106
Agree	43		
Neutral	23		
Disagree	14		
Strongly Disagree	5		
Don't know or not applicable	7		
No Response	1		

(c) the quality of time I spend with my advisor

Strongly Agree	28	Average: 3.94	Count: 104
Agree	39		
Neutral	26		
Disagree	9		
Strongly Disagree	2		
Don't know or not applicable	9		
No Response	1		

G.3. I am satisfied with and interested in my research project.

Strongly Agree	32	Average: 5.25	Count: 101
Agree	44		
Neutral	23		
Disagree	2		
Strongly Disagree	0		
Don't know or not applicable	12		
No Response	1		

H.1. There is a supportive graduate student community in the department.

Strongly Agree	12	Average: 2.48	Count: 111
Agree	51		
Neutral	30		
Disagree	16		
Strongly Disagree	2		
Don't know or not applicable	3		
No Response	0		

H.2. Graduate students in the department are adequately involved in decisions relevant to their education.

Strongly Agree	1	Average: -0.64	Count: 101
Agree	24		
Neutral	43		
Disagree	27		
Strongly Disagree	6		
Don't know or not applicable	13		
No Response	0		

H.3. Faculty in the department work at helping students achieve their goals in research and professional development.

Strongly Agree	7	Average: 2.62	Count: 105
Agree	55		
Neutral	30		
Disagree	12		
Strongly Disagree	1		

Don't know or not applicable	9
No Response	0

H.4. The department provides an environment in which members of underrepresented groups feel comfortable and supported

Strongly Agree	7	Average: 2.10	Count: 81
Agree	31		
Neutral	34		
Disagree	7		
Strongly Disagree	2		
No opinion or not applicable	32		
No Response	1		

H.5. I have adequate time and freedom to pursue interests and activities outside of my academic program.

Strongly Agree	6	Average: -0.76	Count: 112
Agree	39		
Neutral	21		
Disagree	24		
Strongly Disagree	22		
Don't know or not applicable	2		
No Response	0		

H.6. I receive adequate financial support to maintain an acceptable standard of living.

Strongly Agree	8	Average: -0.90	Count: 105
Agree	24		
Neutral	28		
Disagree	31		
Strongly Disagree	14		
Don't know or not applicable	9		
No Response	0		

H.7. I am satisfied with the quality, availability and affordability of housing in the area.

Strongly Agree	2	Average: -3.66	Count: 101
Agree	18		
Neutral	15		
Disagree	36		
Strongly Disagree	30		
Don't know or not applicable	12		
No Response	1		

H.8. I am satisfied with the degree of informal interaction with
(a) other graduate students

Strongly Agree	12	Average: 2.54	Count: 112
Agree	46		
Neutral	41		
Disagree	13		
Strongly Disagree	0		
Don't know or not applicable	2		
No Response	0		

(b) faculty

Strongly Agree	4	Average: 0.72	Count: 111
Agree	43		
Neutral	35		
Disagree	23		
Strongly Disagree	6		
Don't know or not applicable	2		
No Response	1		

I.1. Overall, I am satisfied with the courses in the department.

Strongly Agree	9	Average: 2.89	Count: 114
Agree	63		
Neutral	27		
Disagree	15		
Strongly Disagree	0		
Don't know or not applicable	0		
No Response	0		

I.2. Overall, I am satisfied with my advisor

Strongly Agree	37	Average: 5.28	Count: 106
Agree	44		
Neutral	19		
Disagree	6		
Strongly Disagree	0		
Don't know or not applicable	8		
No Response	0		

I.3. Overall, I am satisfied with the department.

Strongly Agree	11	Average: 3.55	Count: 114
Agree	70		
Neutral	22		
Disagree	11		
Strongly Disagree	0		
Don't know or not applicable	0		
No Response	0		

I.4. Overall, I would recommend this department to prospective graduate students.

Strongly Agree	20	Average: 4.42	Count: 113
Agree	67		
Neutral	19		
Disagree	7		
Strongly Disagree	0		
Don't know or not applicable	1		
No Response	0		

J.1. I am currently

An M.S. Student	7
a Ph D. Student	103
No Response	4

J.2. My major area (choose the best match)

AI	19
Computer Systems	19
Database Systems	7
Software Engineering/Programming Langs.	17
Scientific Computing	4
Algorithms and Computation Theory	11
Visual and Geometric Computing	15
Other	10
No Response	12

J.3. I have been in the graduate program for

less than 1 year	21
1 years	6
2 years	26
3 years	24
4 years	13
5 years	13
6 years or more	5
No Response	6

J.4. (For Ph.D. Students) What is the total number of years you expect to be in the program before receiving your Ph.D. degree?

less than 2 years	2
2 years	2
3 years	6

4 years	15
5 years	46
6 years	22
7 years or more	7
No Response	14

J.5. I am or have been a teaching assistant (TA)

Yes	74
No	32
No Response	8

J.6. I am or have been a research assistant (RA or GRA)

Yes	85
No	20
No Response	9

J.7. My sex

Female	15
Male	88
No Response	11

J.8. My citizenship

US Citizen	34
Permanent US Resident	2
Non-US citizen	68
No Response	10

J.9. (For US citizens only) My ethnic background. (The available choices for ethnicity are derived from existing National Science Foundation surveys.)

African American	0
Asian-American, Pacific Islander	2
Caucasian	32
Chicano/a, Hispanic, Latino/a	2
Native American, Alaskan Native	0
Other	5
No Response	73

J.10. Family life information:
(a) My marital status

Single	62
Married or partnered	43

No Response	9

(b) I have dependent children living with me

Yes	13
No	87
No Response	14

J.11 I have a disability

Yes	1
No	104
No Response	9

Contents

Physical Infrastructure Summary

A.V. Williams Building (Space shown in square feet)

Total Sq. Footage	Adm. Offices	Instructor Offices	Prof Offices	TA Advising	Grad Labs	Spec Res Facilities	Computing Support	OIT Lab	Library	Conf Rooms	Shared Unit Space	Miscel
36,947	2,388	1,438	9,839	462	11,313	3,375	3,000	910	800	1,120	848	1,454

CSIC (Space shown in square feet)

Total Sq. Footage	Classrooms	Computer Lab	Miscel.
15,172	12,798	960	1,414

Contents

Contents

Department Support Structure

The support structure for the Department is composed of 25 staff members, 19 of whom provide administrative support and 6 who are responsible for computer systems and operations. Individual positions and brief function descriptions are outlined below. The Department has returned to its 2000 staffing level due to financial constraints.

Office of the Chair:

Director of Administration: Pat Ipavich, Chief Administrative Officer responsibility for oversight and coordination of the administrative activities of the Department.

Administrative Assistant I: Adelaide Findlay, Receptionist, schedules meeting rooms, purchases department supplies, coordinates the department Distinguished Lecturer Series, provides miscellaneous secretarial support.

Information Specialist: Jordan Landes, serves as the Department Librarian and Newsletter and High School Programming Contest Coordinator.

Business Office:

Assistant Director for Business and Finance: Janice Jastrzebski, responsible for the overall management of the department's financial and business operations.

Business Office Manager: Janet Doherty, responsible for the day-to-day supervision of the Business Office staff, signs payroll journals, final approval for travel requests and PHR actions, monthly review of credit card transactions, and research proposal coordination as needed.

Research Coordinator: JoAnn Simms, responsible for the MD Foundation Account, Research Budgets and Proposal Coordination, and Cost-sharing Accounts.

Coordinator, Payroll: Jodie Gray, responsible for the processing of all appointment information and salary charges to appropriate accounts through the PHR system for department personnel.

Accounting Associates, Grant and Contract Accounting: Cheri Tubman and Kathy MacLeod, balances all research accounts on a monthly basis, prepares financial statements for PI review, verifies the availability of funds prior to expenditures being approved, processes all accounting transactions for designated accounts.

Accounting Associate: Carole Hankins, processes transactions and balances on a monthly basis the State, DRIF, Summer Programs' account, CP Foundation Accounts, and Scholarship/Fellowship Accounts.

Accounting Associate, Benefits and Leave Reports: Sue Blandford, provides benefit information to department personnel, processes faculty leave reports and maintains annual and sick leave balances,

processes corrections to staff leave reports, collects payroll documentation needed for PHR system appointments.

Administrative Assistant I: Felicia Chelliah, verifies the accuracy of Effort Reports, files and performs general office support: (part-time).

Education Office:

Advisor/Consultant, Graduate Program (Faculty position): Gwen Kaye, responsible for graduate and undergraduate class assignments, supervises staff members, coordinates the graduate student application through acceptance process, makes graduate and undergraduate TA assignments, completes reports and responds to information requests from a variety of university and department sources. Works closely with the Director of Graduate Education and advises the Department Chair on a variety of academic matters.

Advisor/Consultant, Undergraduate Program (Faculty position): James Maybury is the primary advisor on academic policies and procedures for all of the Department's undergraduate student population. Interacts with University Administrators and Department faculty members. Compiles information for reports and interacts with the Co-Directors of Undergraduate Education.

Administrative Assistant II, Graduate Program: Fatima Bangura, provides administrative support to the graduate program.

Administrative Assistant II, Undergraduate Program: Julie Buck, provides administrative support to the undergraduate program.

Facilities Management:

Business Services Specialist: Brenda Chick, responsible for the mailroom operation, phone changes, furniture purchases and office moves, enters changes to the card-key system, handles security issues, monitors building maintenance problems, and coordinates parking assignments. Supervises several part-time students. Brenda has continued to perform some administrative responsibilities associated with the Office of the Chair in lieu of the Executive Administrative Assistant position not being filled.

Computer Lab Support:

Director of Computing: Brad Plecs, responsible for maintaining the department's hardware and software computing requirements, manages a UNIX, PC, and Mac based computer environment. Supervises staff members.

Research Programmers: Eric Gurevitz, Matt Katsouros, James Phongsuwan, Geoff Ransom, Derek Yarnell, provides the technical support needed to meet the Department's education, research, and administrative requirements.

Student Assistants: 6-8 hourly paid students

Coordinator: Kim Thangpijaigul, responsible for CALF accounting, computer equipment purchases and their

inventory.

Research Office Support:

Computer Engineer: Steve Kelley, provides programming support to the Database group.

Program Management Specialist I: Cathy Sinex and Tina Madison, both provide administrative support to several faculty members including purchasing, travel coordination, entering information for power-point presentations, coordinating foreign visitor arrangements, monitoring research account statements.

Administrative Assistant I (part-time): Felicia Chelliah, provides secretarial support to three faculty members.

Note: There are several Faculty Research Assistants who provide additional research support to various faculty members. These positions are not classified as staff positions within the University's Personnel Structure.

Contents

Contents

Diversity

Faculty (Instructional, Research, and Professorial Faculty) Fall 2002

Title	# in Category	Gender	EEO Designation
Lecturers	2	F	Caucasian
"	1	M	Asian/Pac. Islander
"	9	M	Caucasian
"	1	M	Hispanic
Instructor	1	F	Caucasian
Adv. Consultant	1	F	Caucasian
"	1	M	Caucasian
Fac Research Asst	4	M	Asian/Pac. Islander
"	1	M	Caucasian
Asst Professor	1	F	Caucasian
"	2	M	Asian/Pac. Islander
"	9	M	Caucasian
Assoc Professor	2	F	Caucasian
"	3	M	Asian/Pac. Islander
"	7	M	Caucasian
Professor	1	F	Caucasian
"	3	M	Asian/Pac. Islander
"	18	M	Caucasian
Emeritus/Dist. Prof.	3	M	Caucasian
"	1	M	Asian/Pac. Islander

Staff (25 Exempt and Non-exempt Positions) Fall 2002

2	F	Af. Am./Black
14	F	Caucasian
2	F	Asian/ Pac. Islander
6	M	Caucasian
1	M	Asian/Pac. Islander

Undergraduate Student Program (based on Fall semester enrollment figures)

	'96	'97	'98	'99	'00	'01	'02
Female	232	292	349	377	378	358	293
Male	1005	1242	1403	1618	1561	1537	1372
Total	1237	1534	1752	1995	1939	1895	1665
Afr Amer	141	185	196	214	214	187	162
Asn Amer	324	429	488	665	621	521	564
Hispanic	37	47	59	77	60	50	55
Native Amer	6	6	4	2	3	5	2
Caucasian	729	867	1005	1037	1041	895	790
Unknown	0	0	0	0	0	237	92

Graduate Student Program (based on Fall semester enrollment figures)

	'99	'00	'01	'02
Female	41	31	30	42
Male	173	223	170	205
Total	214	254	200	247
Afr Amer	7	2	3	3
Asn Amer	NA	7	2	4
Hispanic	8	5	3	4
Native Amer	0	0	1	1
International	158	169	132	165
Caucasian	41	71	59	70

Degrees Awarded during the period July 1, 2001 – June 30. 2002

(By Gender)	BA	MS	PhD
Female	37	6	3
Male	218	20	21
Total	255	26	24
(By Ethnicity)	**BA**	**MS**	**PhD**
Afr Amer	9	0	0
Resident Asian or Pac Islander	89	3	0
Hispanic	4	0	1
Native Amer	0	0	0
Resident White, Non-Hispanic	135	5	4
Other	18	18	19
Total	255	26	24

Contents