1990 Annual Report

Dept. of Computing and Information Sciences

Dr. Virgil Wallentine

March 1, 1990

Table of Contents

Tables

Appendices

Appendix 1.	Baccalareate Requirements
Appendix 2.	Master of Science Requirements
Appendix 3.	Doctor of Philosophy Requirements
Appendix 4.	Teaching Assignments
Appendix 5.	Committee Assignments
Appendix 6.	Faculty Publications
Appendix 7.	Grantsmanship
Appendix 8.	Current Research Programs of the Faculty
Appendix 9.	Professional Activities of the Faculty
Appendix 10.	Summer On Campus Newsbrief
Appendix 11.	Guidelines to PhD Students

I. Preface

This has been a rewarding year for CIS. We hired two new faculty who have enhanced both our teaching and research productivity. The faculty, in general, have increased research productivity, been involved in many professional activities, spread the KSU word across the globe, and been conscientious in transferring knowledge to "budding" new computing scientists. The faculty are generating more research proposals and concentrating on acquiring a national reputation for the Department. We have a young and exciting faculty to match our young and emerging discipline. We are working to make our research and instructional programs relevant. We want to work with other disciplines to enhance their research programs with the computational engineering and science research (computational modeling, computational control, database access, software engineering capacity, etc.) paradigm. Work in this area shows great promise for increased funding through the "high performance computing initiatives" in Congress today. We also want to be involved in the training of MS and PhD students in computational engineering and science. It is projected that there is a need for 1000 PhDs in this area by the year 2000. (See the section on Strategic Plannning for more details). In short, we are excited about our field and its contribution to the University, the State, the Nation, and the World. The world is now a global village and computing and information sciences is a focal point for much of the technology and theory to enable the societies of our world to intermingle with a diversity of cultural, industrial, business, governmental, research, and educational activity. Our faculty understand this societal impact of our technology and want to further its goals.

This has also been a very frustrating year. For the first time in its history, CIS has succumbed to the "survival mentality" that so pervades KSU. In the past, we have always viewed underfunding as something we could overcome. We always had too many students, too little equipment, too few faculty, and (as a result) too little time for research. But we have always heard "lip service" from central administration on how important CIS was to the University. Thus, we continued to overextend, gambling that the University would sooner or later believe that CIS was important, that Computer Science was actually to receive funding to accomplish the mission set forth by the Kansas Board of Regents in naming Computer Science as a central thrust of the University. However, it is becoming more clear to the faculty and students each year that no significant increase in support is imminent. Thus, they are depressed and letting the traditional KSU "survival mentality" set in.

Let us be more specific. While the Department has acquired more than $4 million dollars worth of computing equipment, software, and networking in the past several years, the University has provided less than $150,000 in the past 3 years for upgrade and maintenance of computing labs for CIS. Furthermore, while the Department has been requesting a laboratory fee since 1982 to partially cover these costs, none has been supported by central administration. Our peers receive from $100,000 to $300,000 annually for this purpose. We currently are a faculty of 15, while the national average for PhD-granting Computer Science departments is 20 faculty members and the top 24 Computer Science departments average a faculty size of 28. Our OOE budget is 50% of the OOE of our sister department at KU, while we are essentially the same size in terms of number of students, equipment base, and faculty size. Faculty at the full professor level received 25% less salary than our peers; associate professors' salaries are 15% deficient in comparison with our peers; and assistant professors receive salaries which are 6% below their peers. GTA stipends are approximately 20% below our competition.

But the faculty in CIS are not pessimists; we have grown up in a discipline which changes constantly and we believe in change for the better. Thus, in the section on Strategic Planning, we present a plan to recover from this state of bankruptcy. It takes an investment by the University to make CIS a major thrust of KSU. We want to be a Top 45 department in the country so that we can compete for the increased extramural funding that is predicted to become available from the federal government. Unless we are considered a Top 45, we might as well reverse the trend of the past several years which has seen more than a doubling of extramural funding, increased research productivity by a factor of 4, and improved our computing facilities through extramural grants.

It is time for central administration at KSU to decide if CIS is to be nurtured to enable large payoffs in extramural funding and improved instructional programs. Or, shall we return to the status of 1982, when we were principally a teaching department? We have done remarkably well with extremely low budgets, but we have reached the point of no return in our quest for national stature. We need funding for every aspect of our program. It is time for KSU to do its part. The faculty have done theirs.

II. Instructional Programs

A. Undergraduate Programs

1. Curriculum Development

The Undergraduate Studies Committee is responsible for the health and well-being of all the undergraduate programs - BS/BA in Computer Science and BS/BA in Information Systems. In general, this committee supervises all curricular matters, proposes changes to curricula, interfaces with the College curriculum committee, and interfaces with the Graduate Studies Committee to insure a smooth transition from undergraduate to graduate school.

During the past year the Undergraduate Studies Committee recommended a major restructuring of the CIS 110 and CIS 200/20X courses.

The changes in the CIS 110 course enabled students to have more time in a computer laboratory with an instructor present. The committee recommended that the lecture component be two hours and the third hour be a recitation in a computer laboratory. This would allow actual hands-on experience with instructor assistance and application demonstrations to a smaller group. This change was implemented starting with the fall 1990 semester. Results to date indicate a much improved level of satisfaction and performance with students.

The CIS 200/20X courses were changed to make CIS 200 a three hour course and closely bound to a single teaching language, and to use a graphics approach in teaching the material. Laboratory courses were changed to one hour. The recommendation was made and implemented to require students to take CIS 200 and CIS 203 as a introductory sequence and to use Pascal as the primary teaching language. After completion of the CIS 200/203 courses, the students may then take any of CIS 204 FORTRAN, CIS 206 BASIC, or CIS 208 C. The CIS 204, 206, and 208 language laboratories can be taught at an advanced level, because students should have the common elements of programming from the threshold class. The changes were implemented starting with the fall 1990 semester. Students have indicated a clearer understanding of program design, algorithms, and programming when the lecture is bound to a single language. Because students must take the threshold sequence before taking the BASIC, FORTRAN, or C laboratories, we anticipated a lesser demand for some of those laboratories. Demand for BASIC and FORTRAN have dropped but the demand for C has increased 300%.

The committee reviewed the sequence of theory courses, PHILO 220 Symbolic Logic, MATH 510 Discrete Mathematics, and CIS 570 Theoretical Foundations of Computer Science. Because of the change from CMPSC 370 to CIS 570, some students in that class did not have both PHILO 220 and MATH 510. The committee notified the Department advisors to make efforts to have students take the required prerequisites.

The Undergraduate Studies Committee set the following goals for the school year.

a. Update the statement of focus and goal for each core course.

b. Prepare a proposal for NSF Undergraduate Equipment Grant program. Hankley, Campbell (November 1990)

c. Manage the application for accreditation by Computing Sciences Accreditation Board (CSAB).

d. Make recommendations for faculty supervision of undergraduate service courses.

e. On-going review of existing courses and development of new offerings.

The study of focus and goals for the core courses was concentrated on the CIS 200-300-500 sequence. The problem causing the most concern was what language to employ. Discussion revolved around the need for a language using objects, availability of a good compiler, and one that would fit the programming sophistication of the undergraduate population. The decision was made to continue using Turbo Pascal through the spring 1991 semester, and to further evaluate Modula-2 and Ada.

The committee developed some guidelines for the general content of the CIS 200-300-500 sequence. The 200 course will continue to concentrate on fundamental programming skills, documentation standards, and problem solving skills. The 300 course will introduce common data structures and classes, but students will not be asked to design new abstract data types. The 500 course will require students to analyze alternative data structures, and do considerable analysis of programs for correctness and efficiency.

The committee wrote a proposal for submission to NSF that would create a new course offering. This course would utilize near state-of-the-art computers, software, and networks to lead students from setting up a modern office through utilization of available software packages. The focus would be on the participation of the students in actually doing all the connections, installation of software, scheduling, and similar activities.

The Department has elected to seek accreditation by CSAB for the Computer Science major/BS degree. The first report is due in June 1991 and will require a considerable amount of data collection and consolidation concerning the required courses in the Computer Science major.

A major concern of the Department and the Undergraduate Studies Committee is the quality and continuity of the CIS 110 and CIS 200 service courses. Maarten van Swaay will monitor the CIS 110 class and do development work in the following areas:

a) plan for continuity in teaching
b) set up small recitation/lab sections
c) resolve balance of "tools vs problem solving"
d) incorporate graphic user interface section

Myron Calhoun will monitor CIS 200, William Hankley CIS 300, and Rodney Howell the CIS 500 course and its relationship to our undergraduate theory courses to insure continuity between instructors and appropriate pace in the class.

The committee did preliminary study on the development of an experimental design course that would have C++ as the primary language. The committee discussed what tools might be appropriate for such a course and what relation the course would have to our present CIS 500 and CIS 540/541 sequence.

Items of a miscellaneous nature of concern were:

a. Review of present textbooks in CIS 200, CIS 300, and CIS 500.
b. Coordination with College of Business with our IS curriculum.
c. Letter to Ad Hoc Committee on Common Univ Degree Requirements.

2. Advising System

The Department currently has two undergraduate advisors, one assigned full-time in the Department and devotes as much time as needed to advising. This arrangement has worked well, due to the reduced number of students in computer science as compared to previous years and to the experience level of the advisors.

Undergraduate advising is concentrated in three different events: early enrollments for on-campus students, early enrollments for new and transfer students, and campus visitations. Early enrollments, which occur in April and November, require a concentrated effort by both advisors. Students are normally scheduled by appointment for a 30 minute advising session. The Department maintains a personal folder on each student and the availability of SIS usually enables the session to be efficient and productive. The majority of the undergraduates schedule an appointment, and most are advised during an intensive three week period just prior to the actual early enrollment period. The full-time advisor schedules an additional four-six nights for evening appointments.

Transfer students have early enrollment at the end of the April on-campus early enrollment period and during the summer early enrollment. New freshmen can enroll during the summer period. The summer early enrollment is usually the last three weeks in June, and runs from Tuesday through Saturday. Both advisors are available for the summer enrollments and the advising is either done individually or by group, depending on the number attending. One of the advisors is on call for the Saturday sessions - the number attending is usually available by 10 A.M. Transfer students will usually have their transfer courses evaluated, if not the transcript analyst in Arts & Sciences can give a quick response. New freshmen have a limited number of courses from which to choose, the most critical being the level of mathematics at which to begin.

Campus visitations are sometimes scheduled several days in advance, but more commonly the Department receives just a few hours notice. These students are told about the two majors offered in the Department, information about general requirements, and if time permits, given a short tour of the Department computing facilities. Spring is the busiest time for visitations. The Department will have about 5-10 per week during that time. Each visitation takes about 30 minutes, perhaps a little more if parents accompany the student.

The Department also sends individual letters to students in the Department just before the early enrollment periods to remind them about making an advising appointment. We also send letters to high school and transfer students who have applied to Kansas State. The letter is a general welcome and explains the benefits of a major in Computer Science or Information Systems accompanied by our undergraduate requirements brochure.

Department advising includes some counseling for students having academic problems, dropping classes, and any number of events which can happen on short notice. These are generally handled by the full-time advisor. Department advisors also do informal graduation checks if requested by students.

While no formal survey has been done concerning the success of our undergraduate program, it appears to be successful in terms of informal student comments. Students appear satisfied with the availability of appointment times and the quality of the advice. The number of undergraduates is about 275, which allows the advisors to become well acquainted with most of them.

There is another minor area of advising. There are a number of students who want to work on an additional undergraduate major, and many of these would be better served by taking deficiency courses for a masters, then seeking entry in the masters program. The full-time advisor counsels these students as to their most viable options. There are perhaps one of these counseling sessions per week.

There have been no major problems with the advising program, but we need to continue striving to improve retention. A program to track entering student progress needs to be done as part of the retention effort. This follow-up should be accompanied by some form of exit interview, should a student decide to drop out of school. Informal studies now indicate a fairly high level of drops from the program and low grades appear to be the common symptom, but that cannot be definitely stated, nor can the basic cause be defined. This might require assigning a small amount of additional time to the advising component of the Department. This study might provide some insights as to how to advise new students coming into the program and perhaps some valid screening criteria.

3. Accreditation

For four years we have requested the opportunity to have the Computing Sciences Accreditation Board evaluate our BS program in Computer Science. This year, Interim Dean Kaiser approved our request. Thus, in 1991, we will go through the accreditation process.

B. Graduate Programs

1. Curriculum Development

The Graduate Studies Committee is the primary administrative body for the graduate programs in CIS. This committee processes applications, proposes curriculum changes, evaluates graduate student progress, and recommends graduate students for assistantships. The Graduate Studies Committee is also responsible for administering written preliminary examinations for PhD students and develops guidelines to help graduate students make progress in their coursework and research projects. The current curriculm requirements are listed in Appendices 2 and 3. Guidelines to help PhD students make progress in their research programs are given in Appendix 11.

Planning for the future at KSU, we believe that it is important for the CIS faculty to develop a Master of Software Engineering program. The development of correct and efficient software is essential to the creation of many new engineering and business products. Many of the students in the regular on-campus graduate program will leave KSU and become software engineers. Also, at least 50% of the graduates of the AT&T Summer On Campus program are software engineers. The Graduate Studies Committee is beginning to structure the MSE program using the curriculum developed at the Software Engineering Institute at Carnegie-Mellon as a model. We are also interested in developing a Computer Engineering MS in conjunction with EECE.

The Graduate Studies Committee has been involved in several additional important issues this past year. Specifically the Committee has been discussing new prelim oral guidelines and a possibly new dissertation format. The Committee (after discussing the matter with the faculty) presented to the faculty that we should no longer require an oral as part of the prelim testing; this new prelim procedure was approved by the faculty. The Committee has also discussed whether the department should consider a new dissertation format which would allow a student to have a dissertation consisting of submitted/published papers along with a survey-paper-quality introduction. The Committee's handling of this question was indecisive. Basically, the Committee said that such matters are really in the hands of the individual supervisory committees, but the Committee also noted that some faculty members feel that this (possibly) new procedure is a questionable one.

2. Applications Process

The main activity of the Graduate Studies Committee during this past year has been reviewing applications from students applying to our graduate programs. We continue to see an increase in the number of inquiries and in the number of applications. The Committee have begun a two-step application procedure which allows us to learn about better students sooner and to reject poorer students earlier. However, there are still significant problems in the application process, caused by lack of clerical support for this process.

This past year we processed 785 requests for admission, processed 183 completed applications and admitted 125 graduate students. The quality of these applicants has risen dramatically in the past several years. At present, the average values for GRE verbal, quantitative, and analytical scores are 611, 752, and 698, respectively. The average grade point average of applicants is 3.33. Of those admitted, only 22% (28 of 125) actually enrolled; most of these (74%) were students to whom we offered a GTA. Further, only 50% of the GTA offers we made were accepted. The reason: we do not offer high enough stipends. Even though we raised stipends this past year, we are still 20% below our peers.

We still lose several good applicants because we cannot process them soon enough. For several years we have asked for a classified position in the Department to process applications on a timely basis. We have documented approximately 40 hours per week. Each year it is ignored, even though every other department (that we are aware of) that processes as many applications as CIS has a graduate studies secretary. It seems a shame to reduce the quality of the graduate population because of inequitable administrative treatment.

3. Matriculation of Regular On-Campus Graduate Students

We continue to emphasize our graduate program. The fundamental philosophy is to build a national reputation in graduate education. This builds a strong infrastructure for both research and undergraduate instruction. While we have increased the size and quality of our faculty, the funds available for GTA support from the state has dropped from $342,000 in FY90 to $275,000 in FY92. This is a 25% reduction in support for graduate students. We have seen about that same decline in enrollment. This is certainly not in keeping with Theme 1 of KSUs Strategic Planning process. More details on graduate student enrollment and graduation rates are given in Tables 2 and 7, respectively. Our graduate rate has not dropped as precipitously because we run the AT&T Summer On Campus program, which is funded entirely from extramural sources.

4. A National Stature for KSU CIS in Graduate Education

a. History

Since 1980 the CIS Department has offered a Summer On Campus (SOC) program for AT&T computing professionals. Please see Appendix 10 for a brief history of the program. Each year AT&T funds 50-70 employees with background in computing to come to KSU CIS for 5 weeks. During these 5 weeks the student-employees take two graduate level classes, along with on-campus graduate students. In 5 years, assuming the student-employees do acceptable academic work, they receive an M.S. in Computer Science. We have graduated approximately 80 students from this program and it is extremely successful. We are now ready to enhance this program with electronic delivery of graduate level courses via satellite and interactive graphics to all AT&T sites across the country. This will permit the students to attend only 3 summers on campus and receive the rest of the required courses for the MS through video instruction.

b. National Electronic Outreach from the Plains

We wish to contract with National Technological University (NTU) to teach KSU CIS graduate level courses via satellite. AT&T, along with 75 additional major companies in the U.S., is a corporate member of NTU. The company pays NTU a fee for each course taken by an employee. The member universities are then reimbursed for teaching the courses. In turn, the students can get either NTU credit or credit at the offering university. Our proposal is to provide regular graduate level courses through NTU and enable AT&T SOC students to finish their M.S. degrees in 3 summers on campus and two academic years taking NTU courses offered by KSU CIS. Lionel Baldwin, President of NTU, and Bob Sicora, AT&T Corporate Education Center, both support this move.

Once we have established our credability through NTU, employees at other major corporations can take our courses. This will establish a national prominence for KSU in the major corporations of America. Through this mechanism, any company which has a downlink and is a member of NTU can receive the courses. Ultimately, this may be the right link into the metropolitan areas around Kansas City and Wichita. For example, Allied-Signal Aerospace in Kansas City, AT&T in Kansas City, Boeing in Wichita, NCR in Wichita, and BDM in Leavenworth can downlink these courses. (They are listed in NTUs brochure as being corporate members. There are currently approximately 75 corporate members, including highly influential and respected corporations such as IBM, BELLCORE, General Dynamics, Hewlett-Packard NASA, Eastman Kodak, etc.)

c. Cost Analysis

The total cost to companies is $405 per credit hour. NTU provides marketing, pays for satellite time and collects tuition. NTU then returns $231 per credit hour back to the originating department. If we provide 2 courses per semester with an average enrollment of 25 in each course, we will generate $69,300 per year. Out of this amount, we must pay studio costs for the classroom. I would hope to use the remainder of the proceeds to fund faculty development awards for those faculty members contributing to this program. NTU is moving to compressed video for broadcast. This would reduce the per credit cost and make the program more attractive to companies.

d. University Impact

Once we have established a CIS presence in these corporations, other departments will certainly be interested in providing such instruction. The ultimate impact is that KSU will have a presence within some of the most influential corporations in America. Grants, gifts, influence, and partnerships are much more likely by leveraging this sort of electronic delivery of graduate level courses to help upgrade the technological and scientific expertise of our nation's corporations.

III. Faculty

A. Faculty Composition

The makeup of the faculty is reflective of the two major forces on this department; we work in a young discipline and we are underfunded. We are principally a young faculty with a predominance of assistant professors. We hired two new faculty members this past year, Dr. Olivier Danvy and Dr. Jan Chomicki. This brings our total to 15 faculty members. We still have not reached a critical mass of faculty working in any specific area. Without this critical mass, we cannot be competitive for "big science" grants, nor can we participate in any meaningful way to solving the "grand challenges". We still need at least another eight faculty members whose area of expertise is in one of the following five areas: computational engineering and science, software engineering, programming languages, parallel and distributed systems, and knowledge and data base engineering. In the section on Strategic Planning we carefully enumerate the requisite resources to become a Top 45 department in computing and information sciences.

Because the University has been willing to pay a competitive wage for new PhDs, but unwilling to allocate funds to upgrade the salaries of senior faculty, we are in a severe salary compression situation. At present, full professors earn 25% less than their peers at other Computer Science PhD granting institutions. Associate professors are paid 15% less than their peers. Assistant Professors are within 6% of their peers. One of the fundamental problems at KSU is the unwillingness to compete at the senior faculty levels. It diminishes the instructional and research quality of the entire university and it substantially reduces the capacity of this institution to acquire extramural funding.

B. Faculty Activities

The average faculty member in this department is quite productive. To give an indication of this productivity, an average faculty member:

1. published (or had accepted for publication) 3.5 articles and submitted an additional 3.7 articles,

2. wrote 1.5 proposals for extramural funding (resulting in a department total of approximately $400,000 of extramural funding, 30% of the total department budget),

3. taught 3.1 classes (with an average TEVAL score of 3.7 at the 300/400/500 level, 3.61 at the 600/700 level and 4.0 at the 800/900 level),

4. was major professor for 5 graduate students,

5. served on 2.5 departmental and university committees, and

6. was involved with at least 20 other activities such as prelim development and grading, reviews for outside agencies, talks at other universities, professional society service, supervision of GTAs, facilities acquisition, participation in readings classes, guiding graduate student seminars and projects, membership on supervisory committees, advising student groups, advising students, etc.

Details on faculty activity are contained in appendices 4, 5, 6, 7, 8, and 9. In these appendicies we have enumerated faculty contributions in research, committee service, grantsmanship, teaching, and professional service.

IV. Research

A. General Research Areas

Research activities in this department are broadly categorized into the following general areas: programming languages, software engineering, data base systems, computational engineering and science, computing ethics, and distributed and parallel systems. Appendix 8 contains more detail on each research project.

Programming Languages

Dr. David Schmidt works in the area of programming language semantics. He is currently working with Susan Even on action semantics, which is a high level, readable, beginner's notation for defining semantics of programming languages. He is working with Masaaki Mizuno on a denotational semantics-based correctness proof of security flow control algorithms. He is working with Kyung-Goo Doh on "well-formed" programming languages in the areas of static typing, binding in block-structured languages, and block-structuredness of storage cells. Adrian fiech is a student working with Schmidt in action semantics so that this model can describe polymorphic programming languages like ML. Karoline Malmkjaer is working in a related area to Schmidt. She is working on defining methods for analyzing partial evaluator programs. Dr. Olivier Danvy is working with Banerjee to design an orthogonal Algol-like programming language. Dr. Danvy also works on improving the effectiveness and efficiency of partial evaluation with Charles Consel of Yale University, on models of control with Andrezej Filinski of Carnegie-Mellon, and on the essence of partial evaluation with Karline Malmkjaer. Dr. Austin Melton is working on understanding and developing categorical programming languages and on understanding data types via a categorical construction called dialgebras with Hans Dybjkjaer of DIKU. He is working with Bill Young on programming languages. He is working with Dr. George Strecker and Bernd Schroeder, from Mathematics on Lagois connections properties in applications of computer science. Dr. Jan Chomicki works in the area of logic programming.

Software Engineering

Dr. William Hankley works in the area of temporal specifications for software systems and in the general areas of specification and verification. Dr. Melton is working on developing constructive software measurement with the hopes of producing a major re-evaluation in how software metrics are designed and tested. Dr. David Gustafson is working in the areas of software measures, formal ways of comparing software testing methods, formal models of software development (with Eric Byrne), and software reliability (with Dr. Sallie Keller-McNulty in Statistics).

Data Base Systems

Dr. Elizabeth Unger works in the areas of Data Security and Integrity with Sallie McNulty and Lien Harn (from UMKC). Dr. Austin Melton is working with Dr. Unger and Dennis Ng on complex data objects. He is also working with Dr. Sujeet Shenoi of Tulsa University on a generalized database model called the equivalence-class (partition) relational database model. Dr. Jan Chomicki works in the area of deductive data bases, novel query languages, query processing methods, and finite representation of infinite query answers; he also works in the area of database integrity with emphasis on transition constraints, temporal logic active databases, and triggers. Dr. Zamfir-Bleyberg is working on object-oriented databases.

Parallel and Distributed Systems

Dr. Rodney Howell is working in the area of hard real-time scheduling. He also works in the area of problems associated with petri nets. Dr. Hankley is working on temporal specification of Ada semantics. Dr. Unger is working with Dr. McNulty on active data elements. Dr. Mizuno is working on secure information flow in distributed systems (with Schmidt), recovery in distributed systems, and distributed mutual exclusion algorithms (with Mitch Nielsen). Dr. Unger is working on office information systems. Dr. Zamfir-Bleyberg is developing an entity-relationship algebra, a formal model of concurrency. Dr. Ravindran is developing high performance algorithms for ISDN switches and fault-tolerant remote procedure calls. Finally, Dr. Wallentine is working in the area of parallel discrete event simulation. He is working with Jim Butler on a time-space parallel discrete event simulation language model with applications to more general parallel applications, with YuFeng Li on simulators of parallel discrete event simulators, and with Al Briner on temporal locality in hybrid models of conservative and optimistic parallel discrete event simulators.

Computational Engineering and Science

Dr. Maria Zamfir-Bleyberg is working with Dr. Isenhour in Chemistry to apply artificial intelligence and concurrency control mechanisms to control of analytical chemistry robots. Dr. Rodney Howell is working with P. Krishnaswami from mechanical engineering and George Strecker from math in the area of process planning for machine shops. Dr. Zamfir-Bleyberg is also working on the application of neural networks for the detection and classification of various grain features, with Inna Zayas, USDA Grain Marketing Research Laboratory.

Computing Ethics

Dr. Maarten Van Swaay works in the area of computing ethics. He has published several papers on the responsibility computer scientists must exercise in developing software which is correct, efficient, and effective. Furthermore, he writes about where computing ethics guidelines can or cannot be applied, and where the rules stop and judgement begins.

Extramural Funding

In this past year we have had more than $200,000 in research and educational funding active within the department. Additionally, we have written four equipment grant proposals, 17 research and development grant proposals, and one Presidential Young Investigator proposal. Appendix 7 contains details about our grantsmanship activities. We expect to improve on this area in the next several years, but only if we have a larger faculty so that we can cover both research and teaching duties.

B. Seminar Series

It is essential that research be conducted within the context and knowledge base of our global village. Thus, we have a seminar series which attempts to bring to the plains (KSU) distinguished speakers who can appraise us of the state of our discipline worldwide. A listing of these speakers is presented here.

The department seminar series was very active in the past year. We had a wide variety of speakers.

Our first speaker of the Spring 1990 semester was Guo Qiang Zhang from the Computer Science Department at the University of Georgia. He spoke on February 1 on "Mu-Calculus of Domain Logic."

On February 20, Dr. James Tomayko from the Software Engineering Institute at Carnegie-Mellon spoke about "NASA's Manned Spacecraft Computers."

Dr. K. Ravindran, a KSU assistant professor, described "A Flexible Communication for Distributed Applications" on March 1.

Dr. Vasant Shambhogue, from Cornell University, spoke to us on March 27 on "The Expressiveness of Indeterminate Dataflow Primitives."

On March 29, Dr. Rao Vempaty from the University of Texas at Austin gave a seminar on the topic of "Efficient Parallel Algorithms for State-Space Search."

Dr. Tsung Kuo from SUNY at Stony Brook presented a seminar on "Lazy Functional Programming and Strictness Analysis" on April 2.

Our next speaker was Dr. William Winsborough, from the University of Chicago. His talked was presented on April 3 and entitled "Analysis of Shared Data Structures in Logi Programs for Copy Avoidance."

Dr. Jan Chomicki, University of North Carolina at Chapel Hill, presented the next seminar on April 6, entitled "Finite Representation of Infinite Query Answers."

Dr. Olivier Danvy, KSU visiting professor, presented a seminar on April 9 entitled "The Abstraction and Instatiation of String Matching Programs" and presented a second seminar on April 26 on "Abstracting Control."

One of our PhD graduate students, Richard Courtney, presented a seminar on May 1 entitled "A Formalism for Transformations of Hierarchy Diagrams."

The last seminar of the Spring semester was presented by Dr. Michael Wick of Washington State University. His seminar, on May 11 was entitled "Reconstructive Expert System Explanation."

We had only one seminar speaker in the Summer of 1990. Dr. Christof Ebert, University of Stuttgart, presented a seminar entitled "Visualization Techniques for Analyzing and Measuring Design Decisions" on July 25.

Our first speaker of the fall semester 1990 was Dr. Charles Consel from Yale University. His talk on Thursday, September 27th, was titled "Semantics-Directed Generation of a Prolog Compiler".

The next talk was titled, "Non-Standard Clustering Algorithms" and was given on Wednesday, October 3, 1990 by Dr. James C. Bezdek, who is the Nystul Professor of Computer Science at the University of West Florida.

The third talk was given by Susan Even, a PhD Candidate in the department on the topic, "Type Inference for Action Semantics" on Thursday, October 25, 1990.

Dr. Andrzej Filinski of the School of Computer Science at Carnegie-Mellon University talked on "Declarative Continuations and Categorical Duality" on November 2, 1990.

On November 14th, Dr. Radia Perlman of Digital Equipment Corporation talked on " Calculating a Safe Route in a Computer Network Despite Traitorous Advisors".

Dr. Andrzej Ciepielewski of the University of Iowa, Iowa City presented " Parallel Implementations of Prolog: How to Map Dynamic Trees To Multiple Processors Tuesday, November 20.

The last talk in 1990 was presented by our own faculty member, Dr. Olivier Danvy, who talked on "Partial Evaluation in Parallel" on November 29th.

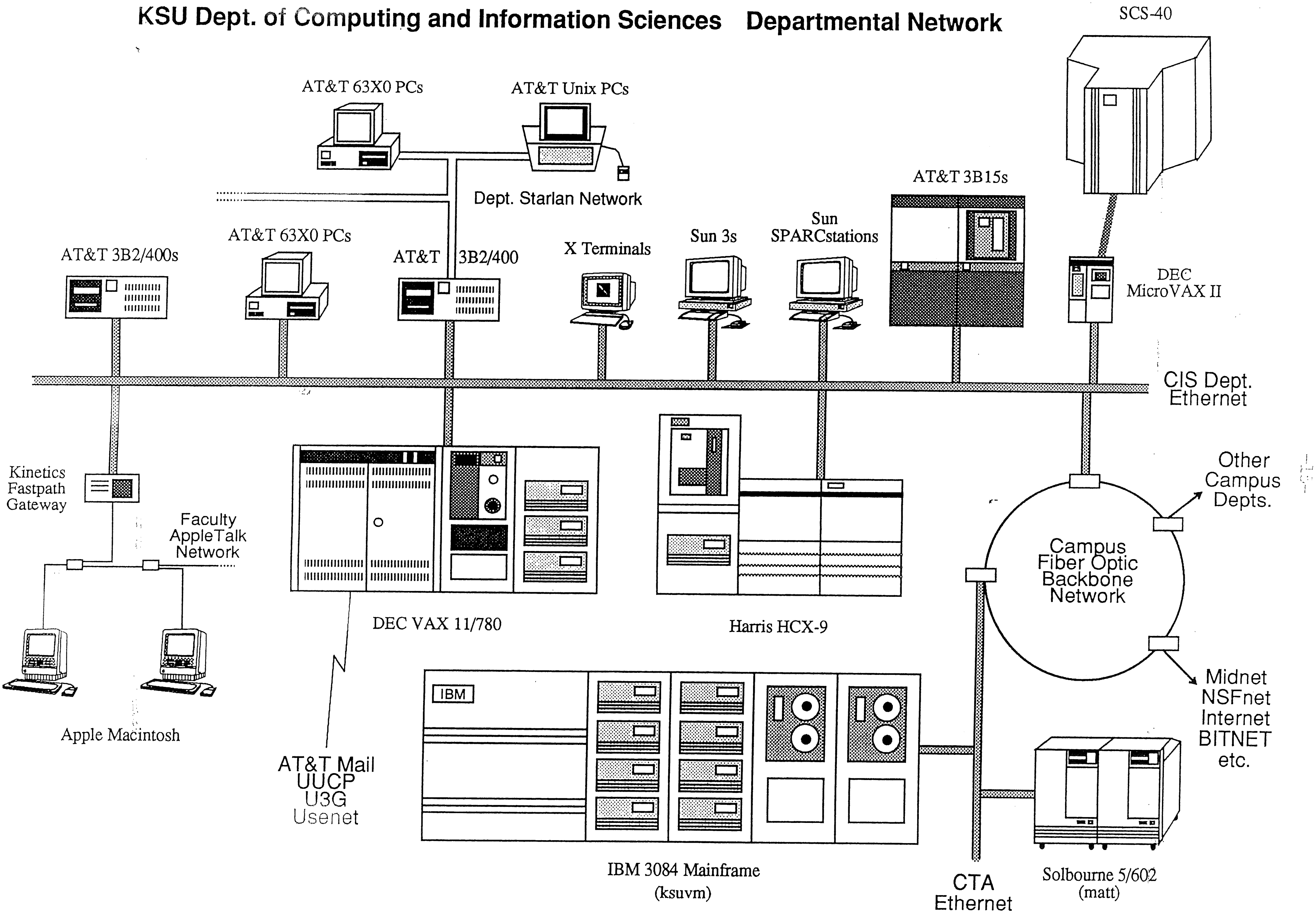
V. Computing Infrastructure

A. Current Environment

Essential to all scientific and engineering disciplines are the tools which promote progress in research, development, and instruction. In Computing and Information Sciences, it is essential that students have state of the art software and hardware laboratories. This is a rapidly evolving technology and lab work on obsolete equipment is nearly useless. Our research paradigms include theory, where we prove theorems about computing objects, engineering, where we design and validate the operation of real computing systems, and experimentation, where we experiment with models of computing and communications systems to test both correctness and efficiency. Our research and instructional capabilities are directly related to the quality of our computing infrastructure - facilities, maintenance, and personnel.

Our present environment consists of approximately 90% obsolete equipment. Table 6 contains a detailed listing of the computing facilities in the Department and Figure V.1. contains a picturial representation of the equipment and its internetworking. We have approximately 100 PCs, 15 minicomputers, 4 mainframes, 1 mini-supercomputer, 50 terminals, 15 X-terminals, 15 workstations, 300 software packages, and 5 networks. This sounds impressive until you realize that most of this equipment is obsolete. Furthermore, much of the equipment is not operational. At present, it would

KSU Dept. of Computing and Information Sciences Departmental Network



take approximately $30,000 to repair this equipment. The only state-of-the-art equipment we have are the workstations and the X-terminals.

B. Deficiencies

In the undergraduate curriculum, we need 25 X-terminals attached to a multi-processor file- and cycle-server. We also need a multitude of software in the areas of software engineering, programming languages, computer networks, operating systems, data base systems, fourth generation languages, and graphics. At the graduate level, we need 10 workstations (Sun Sparc 2) with a multiprocessor file server. In research, we need a large parallel processor for research in parallel algorithms. No self-respecting computing science department can do without this type of computational power. To support interdisciplinary research we need high-resolution color graphics systems for scientific visualization. While this department has acquired most of its instructional and and research equipment, it is now necessary for the University to support the acquisition of this needed computing infrastructure.

C. Grantsmanship

As stated earlier, this department has acquired more than $4 million dollars worth of computing facilities through extramural grants and contracts. That is 90% of our computing capacity. This past year we have been diligent is seeking such support, but all our proposals have been rejected. A proposal was submitted to AT&T for a multiprocessor for graduate education and research ($800,000). It was rejected with no reasons. A proposal was submitted to Hewlett-Packard, jointly with Engineering, which requested equipment and software ($1.5 million) to support undergraduate instruction in Computing and Information Science and in Engineering. It was rejected with no reasons. A proposal was submitted to NSF for support of faculty and graduate student workstations and file servers and support personnel ($1,365,691). It was rejected because we do not have a critical mass of senior computer scientists to warrant a major investment by NSF. Finally, a proposal was submitted to NSF for support of undergraduate instruction. It is pending. One of the fundamental problems we face is lack of state funding to bring us up to a level to be competitive with the rest of the academic world. More detail is provided in the section on Strategic Planning.

VI. Strategic Planning - Top 45

A. Abstract

In this section we present the rationale for funding Computing and Information Sciences at KSU at a level commensurate with the Board of Regents Mission Statement for the University. That is, we want to make computer science a major thrust of the University in fact, not just on paper. The tangible rewards that are possible if CIS is given a central focus within the University's strategic plans are listed below.

1) Federal funding is projected to double in the next five years in computing sciences and engineering. If we are funded at a Top 45 level, there is the potential to generate two million dollars per year in outside funding.
2) An increase in the number of CIS faculty will also enable CIS to improve the quality of our instructional programs and meet the demands implied by the Common Undergraduate Degree Requirements. It will also permit us to extend our graduate program classrooms to a nationwide industrial clientele.
3) Additional faculty in computational science and engineering, specifically in parallel systems and visualization, will permit the building of research partnerships of CIS faculty and computing-intensive research faculty in other disciplines to leverage the power of the computer to be more competitive for the "grand challenge" projects set forth in the High Performance Computing Initiative.

While the cost:reward ratio to attain these goals is not high, we are requesting a significant budget increase. The requisite resources to achieve these goals are:

1) six new faculty in "core" computing sciences;
2) at least two new faculty in computational science and engineering, particularly in parallel computing and visualization; and

3) a computing infrastructure which supports computing-intensive research.

In this document we enumerate the costs of becoming a Top 45 and the rewards that accrue from such an investment by the University.

B. Introduction and Overview

1. History and Objectives

The Department of Computing and Information Sciences at Kansas State University is younger than Monday Night Football and was created after the landing of a man on the moon; it was developed in response to an overwhelming need to enable industries to be competitive, to empower other disciplines with computing power, and to provide knowledge workers for the 21st century. We have been very successful in producing significant numbers of graduates from both our undergraduate and graduate degree programs, increasing our research productivity and providing our own computer labs. However, we have never been allocated reasonable resources to achieve our teaching and research potential. In this proposal, we are asking for the resources to accept the challenge of the Kansas Board of Regents, given to this department in the Mission Statement for KSU; and that challenge is to be a major thrust of the University. Specifically, increasing the number of faculty, improving the stipends for graduate students, and enhancing the research and instructional laboratory environment will substantially increase the size of the graduate program, increase extramural funding, increase the publication rate in scholarly journals, and improve the quality of the undergraduate programs.

2. Goal

Our goal is to become a Top 45 Department of Computing and Information Sciences in the US by 1995-96. There are several reasons for KSU to consider this a top priority.

a. The Board of Regents of Kansas has mandated in its Mission Statement for KSU that Computer Science be a major thrust of the University [Boa86].

b. The federal pool of extramural funding for computing research is predicted to double (from 300 to 600 million per year) in the next five years and a major portion of this funding is to be earmarked for the Top 45 computing science and engineering departments in the country [Won90]. It is our belief that the potential for funding from industrial sources is also increasing.

c. There is a nationwide concern about the lack of human resources in computing and information sciences to enable the nation's information industry. Eugene Wong, Associate Director of the White House Office of Science and Technology, has said that these human resources are the primary factor in enabling the US to maintain leadership in information technology and that information technology affects 2/3 of our GNP [Won90].

d. Research in many disciplines which depend on advances in computing will be disadvantaged at KSU if interdisciplinary research in computational science and engineering is not a strong part of the research infrastructure of KSU.

e. Enhancing the CIS program is in exact agreement with KSU's criteria for prioritizing programs.

f. Achieving a critical mass of CIS research faculty and infrastructure is consistent with the strategic planning themes of KSU.

Let us be more specific.

3. Kansas Board of Regents Mission Statement

In the mission statement for KSU the Board of Regents states, "The major thrust of the institution is guided by its land grant tradition, embodied in agriculture, architecture, business, **computer science,** engineering, veterinary medicine and human ecology/home economics, and those disciplines necessary for the support of these fields: the natural and physical sciences, mathematics and the social sciences." We are a major thrust of KSU and we are asking to be funded sufficiently to meet this challenge.

C. Funding Potential

There is a computing initiative before congress (Gore Bill - S. 1067 National High-Performance Computing Technology Act of 1989, Johnston Bill - S. 1976 A Bill to Provide for Continued United States Leadership in High-Performance Computing, Walgren and others - H.R. 3131) which could possibly double the amount of funding for computing research in the next five years. At the 1990 Snowbird conference, the bi-annual meeting of the Heads (Chairs) of the PhD-granting Computing and Information Sciences Departments in the U.S. and Canada, this high performance computing initiative was discussed with representatives from industry and all the major granting agencies. The attendees who were most knowledgeable of the high performance computing initiative and most influential in structuring the initiative were Eugene Wong - Associate Director, White House Office of Science and Technology Policy, Erich Bloch - Director of NSF, Bill Wulf - Director of the Computing and Information Science and Engineering Directorate of NSF, Barry Boehm - Director of DARPA/ISTO (DOD), and Andre van Tilborg - CS Director of ONR (DOD). The message was clear: the High-Performance Computing Initiative will attempt to bring 35 additional departments of CIS up to the level of the current top ten research departments in CIS in the U.S. Any department below the top 45 research departments would clearly be in a (possibly long-term) disadvantaged position. While there was a lively discussion about "big science" versus "small science", most agreed (during the coffee breaks) that there would be a top 45 and then the rest. This was specifically stated by Eugene Wong. It is clear that if we are to be a major source of research funding for KSU, we must be a top 45 department. Currently we are not.

In a related document [Hop89], the NSF Advisory Committee for Computer Research proposes "That a major initiative in software engineering research be initiated ... [and] ...a new program of research for medium-size, multiple-investigator projects be initiated. Such Projects would typically be funded at $200,000 to $800,000 per year." This tells us that being a top 45 department will have significant rewards.

D. Dearth of Human Resources in Computing

In his address to the the 1990 Snowbird conference, Eugene Wong also stated that there is a "decline in the supply of human resources in the computer science educational pipeline". He later indicated (and we paraphrase) that the federal government must have a hand in improving this supply of computing professionals because the competitiveness of the US in the world economy depends heavily on information technology, and computing is the "driving technology" of the industry. We have also seen this dearth of computing professionals in our connections with industry that recruit at KSU in CIS.

E. Interdisciplinary Research in Computational Science and Engineering

Increasingly, science and engineering researchers are turning to computational models of physical, chemical, and biological systems to conduct their basic research. Scientists and engineers are also more dependent on computing technology and processes to implement prototypes and control experiments. A partnership of scientists and engineers at KSU who are actively involved in research with CIS faculty who are experts in parallel and distributed algorithms, scientific visualization, artificial intelligence, and software engineering would significantly increase the potential of success in research funding. This strategy for interdisciplinary research teams in computational science and engineering is documented in the High Performance Computing Initiative which indicates the essential nature of computing in tackling the "grand challenges" of science and documents a national purpose of maintaining international superiority in computing and computing technology.

F. Compliance with KSU Planning Criteria

Clearly, improving the quality of research and instruction in CIS coincides perfectly with the mission of KSU, as embodied in the Board of Regents' statement. Second, demand for our degree programs has already been documented. The quality of our programs is clear from the demand for our graduates from all of our programs and our own determination that our BS curriculum in Computer Science qualifies for accreditation by the Computer Science Accreditation

Board. Furthermore, we are the only program in Kansas which brings Computer Science programs and Information Systems programs together within the same department. Finally, the instructional programs provide courses central to the mission of all KSU departments and research central to the computational science and engineering research paradigm.

G. Relationship to Planning Themes

Enhancement of the graduate education and basic research programs of CIS is central to meeting the Regents' challenge. Strengthening our research programs will also improve the quality and quantity of undergraduate education in this high demand area. Critically needed CIS faculty will help strengthen the service teaching programs mandated by the Planning Charge to the College of Arts and Sciences [KSU89]. Electronic delivery of graduate coursework will extend the influence of KSU to industries nationwide. Finally, a cooperative research and graduate education program with the European computing sciences community will enhance the international stature of KSU.

H. The Plan: Investments and Rewards

1. What Needs to be Done

The CIS Department at KSU is not a top 45 department; to enter this elite group, and thereby enable us to compete for significant multi-investigator large grant extramural funds, we must:

a. increase the number of research faculty in areas with potential for funding,

b. hire established researchers with good "track records",

c. create critical mass research groups,

d. focus on excellence in a few research areas,

e. provide the research infrastructure - computing facilities and operating budget - to entice new computing sciences researchers to KSU,

f. increase the number of quality teaching faculty,

g. insure the quality and quantity of comprehensive teaching laboratories, and

i. extend the KSU CIS graduate classroom, via electronic delivery, to a nationwide industrial clientele.

2. Departmental Impact of a Top 45

We are currently a department of 14 tenure track faculty, generating about $30,000 per year per faculty member in extramural funding. If we are to understand the economic impact of creating a critical mass of researchers in CIS, let's look at the configuration of the current Top 25 departments in the US, as reported in [NRC82]. While numbers in the rapidly changing academic computing field are few, we do know that, according to [Gri90] and [Gri86], the Top 25 are substantially larger in faculty numbers and extramural funding than the remaining 75+ programs. The Top 25 have an average faculty size of 28 and average $140,000 per year per faculty member in extramural funding from the federal government. (Industry funding is unreported, but is believed to be a significant percentage of the federal funding.) The remaining departments (below 25) average approximately $20,000 per year per faculty member. It seems clear that a critical mass of faculty in research contributes to increased extramural funding. If KSU CIS were raised to the level of a current Top 25 in faculty size and extramural funding, we would have the potential for generating an additional two million dollars of extramural funding per year (in 1986 dollars). Including consideration of the High Performance Computing Initiative, this potentially produces double the 1986 dollars - four million dollars per year.

The increased visibility and reputation of a top 25 department also improves the quality of graduate and undergraduate students. There is a significant national concern (again using the 1990 Snowbird Conference and industry demand as indicators) for the dearth of human resources in computing. Based on the number of qualified students we cannot admit because of lack of funding and faculty, we feel we can attract additional highly qualified students.

3. Needs of a Top 45

In the previous section, we used the Top 25 (in 1982) as a benchmark against which to judge the rewards of a high quality Department of Computing and Information Sciences. We feel it is a good benchmark for being a top 45 department. The rewards of being a Top 45 department are now clear; let us make sure that the costs are also very clear. First, we will need an additional 8 faculty members to reach a critical mass of 22 tenure-track research faculty. While this is substantially below the average size (28) of a top 25 department, it is our belief that there are still a significant number of courses which can be taught by GTAs and thereby reduce the teaching load on faculty and this will be an investment of approximately $600,000 per year in salaries. A computing infrastructure must be established to entice quality faculty to KSU. A large parallel processor, high-resolution graphics facilities, and faculty workstations are an investment of approximately $800,000. Maintenance and management of these facilities will require approximately $100,000 per year, in addition to our current $100,000 per year deficit in maintenance of computing equipment and software. However, if we do our faculty hiring, promotion, and development in a responsible way, two to four million dollars per year, plus national stature, are the rewards.

I. Research Focus

1. Background

If we are to be competitive in acquiring funding from such agencies as NSF, according to their Notice No. 107 [NSF89] we must "contribute to the education and the development of human resources in science and engineering at the postdoctoral, graduate, and undergraduate levels." Additionally, NSF states that "Evaluation of scientific productivity must emphasize quality of published work rather than quantity." Our interpretation of these statements is that we must publish in first-rate journals, develop high quality graduate and post-doctoral students, and develop a critical mass of graduate students with which principal investigators can work. Currently, we do not have a critical mass of faculty in any area; if we lose one person, an entire area may be crippled. It is impossible to compete for "medium and big science" grants without such a critical mass. We propose to strengthen current areas of expertise with additional faculty in the following three broad areas: programming languages, parallel systems, and software engineering. It is important to note the specific sub-areas which have been included in these broad areas. Under software engineering, [Hop89] included data systems, operating sytems, software development, software maintenance and testing, software design, and software environments. Distributed computing was included in parallel systems as well. We have chosen these three broad areas for several reasons. First, we already have the beginnings of research groups in these three areas. Second, the NSF Advisory Committee for Computer Research has recommended that these areas be among those chosen for high priority in future research funding [Hop89]. Third, these research areas are mutually supportive of one another, that is, several faculty can work in research that integrates these areas.

2. Core Computing Sciences Research Focus

Let us now provide a more specific focus. Our strongest research program is in programming languages. At a more detailed level, our strengths in denotational semantics, applicative programming, logic programming, and category theory have given us the beginnings of a national and international reputation. By adding two more faculty positions in programming languages, we feel confident that we can establish this program as among the better programs in the nation. Presently, several faculty have strong connections with programming language research groups abroad at Copenhagen, Edinburgh, Glasgow, and Rennes (France). Research visits and postdoctoral students have been exchanged, and NSF has granted funding to continue the exchanges. We want to expand this human resources pipeline to include PhD students (one this year already). Additionally, there is a potential for multi-million dollar grant cooperation through the European equivalent of NSF, called ESPRIT. KSU (along with Yale and Northeastern) is listed as a potential participant in new NSF/DARPA/ESPRIT initiative [Hud90], but we need additional high quality faculty in programming languages to attract this funding.

In parallel systems and software engineering, we have faculty who work in data systems, distributed systems, distributed algorithms, software metrics, software specification and verification, software debugging, and concurrent systems. However, we do not have a critical mass in any of these areas and no one that works in parallel algorithms and systems. Our goal is to build a critical mass of faculty researchers who can provide national leadership in "Software for Parallel Systems". This means that we must add two software engineering faculty and two faculty in parallel algorithms. More optimally, we need three in each of the areas; however, it is intended that the two positions in programming languages will have some expertise in software engineering or parallel and distributed language structures, and thereby strengthen our ability to conduct significant research in software for parallel systems.

At present, our faculty is dominated by bright, young, hard-working faculty who hold great potential. However, in order to upgrade this department in time to acquire money from the High Performance Computing Initiative, we must have senior faculty who can compete on the national level. Therefore, at least three of these new faculty positions must be at a senior level.

3. Interdisciplinary Research

The High Performance Computing Initiative [Hol90] clearly states that fundamental research in computing and information sciences is essential to the well-being and economic development of the US. It also gives computer scientists an additional task. Computing scientists are urged integrate their research into the national purpose of competitiveness in information technology and embed their research goals into some of the "grand challenges" of other disciplines. Global weather modeling, mapping the human genome, improving the product cycle, and development of computational scientists and engineers are given as examples of such grand challenges. Thus, it is essential that we create a critical mass in computing research by developing interdisciplinary computing research teams with members from CIS and other disciplines. With this goal in mind, CIS submitted the RIACT pre-proposal for strategic planning in the Fall of 1989 [Wal89]. The goal of RIACT is to build on the strengths of CIS and other disciplines that have computational research problems. For example, linguistics research in speech and English can benefit from knowledge in programming language semantics. Parallel and distributed algorithms in CIS provide a powerful computational tool to many of the scientists and engineers in solving their computationally intensive simulation problems. Deductive data bases, data base integrity and security, and object-oriented data bases can be significant factors in solving problems in computational biology, geographical data bases, office automation, flexible manufacturing, etc. Parallel algorithms can help to solve problems in computation science and engineering, such as computational physics, computational chemistry, parallel simulation, circuit simulation, real-time algorithms, and numerical computing. Software specification, verification, and implementation can improve productivity in research areas which must build software systems. Theorem provers in CIS can help mathematicians solve some humanly intractable proof problems.

Once CIS has a critical mass of researchers in the "core" computing sciences, described above, this interdisciplinary research can proceed. The fundamental areas of software engineering, data bases, programming languages, and parallel systems can provide the underpinnings to programs within RIACT for computational science and engineering. Joint projects conducted by faculty from departments such as math, stat, physics, chemistry, biology, bio-chemistry, engineering, and others are obvious.

There is, however, one obvious deficiency in our ability to conduct this type of research. It is essential that we build an expertise in visualization. In scientific visualization the scientist or engineer is enabled by high-resolutions color graphics to use his/her image recognition senses and faculties to understand a physical, chemical, or biological process. To promote this scientific method, two positions in graphics and visualization must be allocated to CIS. With at least one of these two faculty being a senior person, early success in obtaining extramural funding for such projects seems promising because of the strength of the science and engineering programs at KSU.

J. Instructional Focus

1. Enhanced Quality

With the increase in number of quality faculty, instructional programs will also be improved. We must concentrate on presenting quality programs at all levels. Without quality undergraduate programs, our graduate programs and research programs will be disadvantaged because of ill-prepared students. The curricula will not be changed substantially in core computer science areas. However, with the addition of visualization faculty, our ability to train students as computational scientists and engineers will be established.

2. Instructional Infrastructure

It is essential that we install workstations for both the graduate and undergraduate students. Without this laboratory environment, students cannot develop their engineering and experimentation skills or use current research software. In the past we have received lab computers from industrial grants, but because KSU was unable to provide maintenance, industry is reluctant to continue the practice.

It is essential that we improve graduate student stipends. Our current offers are 25% below the offers of our peers. With higher stipends we can increase the number of quality, research-oriented students who accept GTA and GRA positions. We receive more than 1000 inquiries a year and process more than 400 applications for graduate school, admitting 20-30, of which 10 to 15 accept GTA or GRA offers. With additional faculty, better stipends, and better equipment, we could enroll an additional 30 graduate students.

With additional faculty, we can reduce our student to faculty ratio, and qualify our BS in Computer Science for national accreditation.

At the request of AT&T, we propose to offer MS coursework electronically through National Technological University (NTU). We have a current demand for such courses: in addition to their annual summer school attendance, our AT&T Summer On Campus students must receive MS coursework at their business locations (30 sites across the nation). This initial experience with electronic delivery of courses will open the doors to additional industrial clients for our graduate level courses. Because many computer and computing-intensive industries, such as IBM, Xerox, AT&T, and DEC are members of NTU, we can reach into industry throughout the nation. Specifically, we can and should reach into the industrial areas of Kansas City and Wichita to be supportive of Kansas economic development.

Responding to the increased demand for graduate level degrees in computing technology and information management by nationwide industries like AT&T and Kansas industries such as Boeing, United Telecom, and the insurance companies, we propose to develop a new Master's degree program - Master of Software Engineering (MSE). The model for the MSE is provided by the Software Engineering Institute at Carnegie-Mellon.

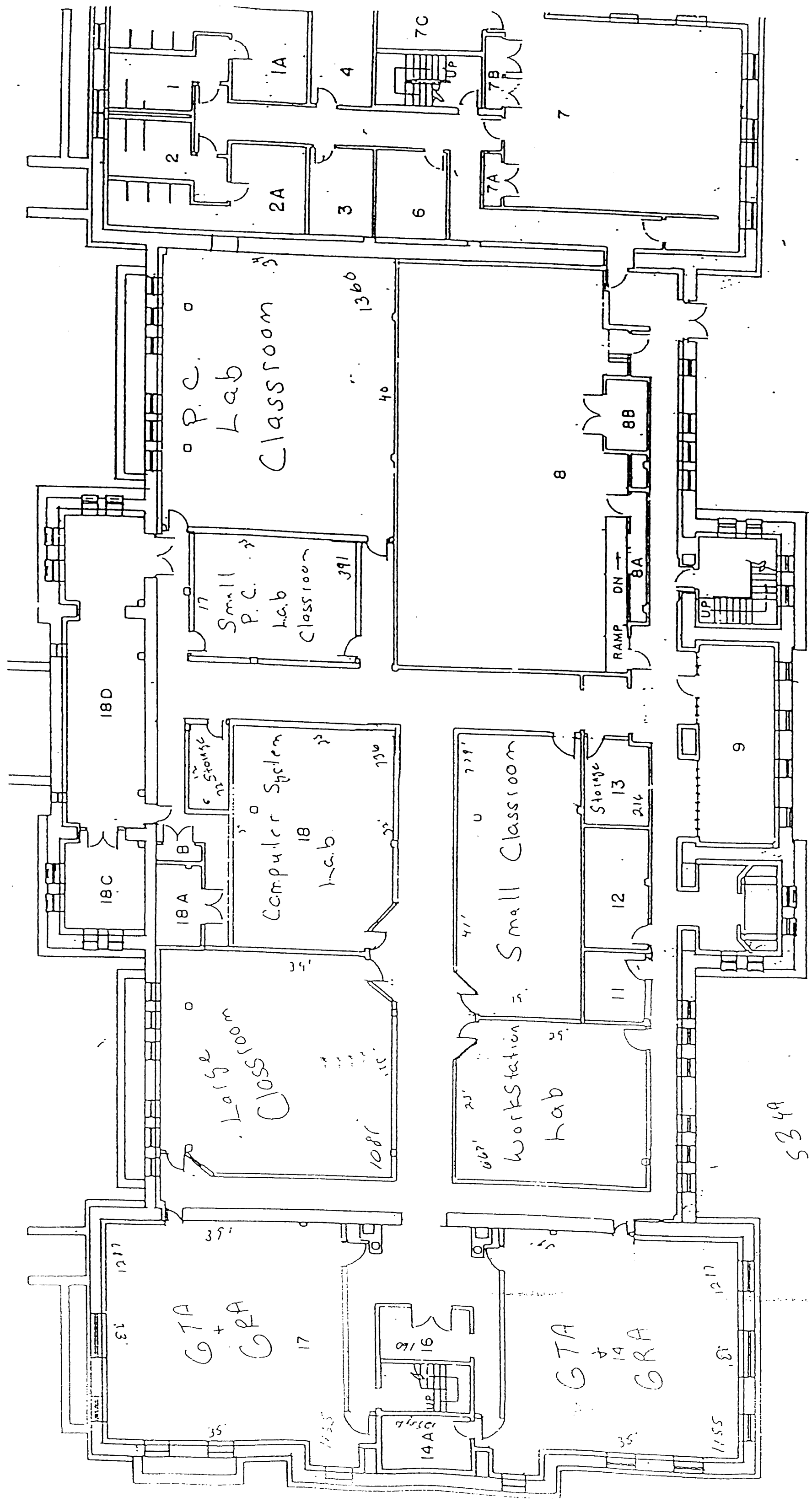
Support for improving the quality of the Joint PhD program with the University of Kansas is desirable. We wish to establish a video link between the two departments to enable us to teach joint PhD level courses, conduct research seminars, conduct collaborative research projects, and interact on economic development projects. KTEC has expressed interest in this type of cooperation.

3. Space Needs

The Department of Computing and Information Sciences has high quality space in Nichols Hall. The State of Kansas has provided a very nice working environment in an aesthetic and sentimental setting. However, we are a growing department and the current separation of labs and staff in both Nichols and Fairchild causes difficulties in management, budget, and morale. Our goal is to consolidate all of our space in Nichols. We also need new space to accommodate class-labs for teaching "hands-on" and experimental computing and information technology. An attachment to this document contains a floor plan of the space in the Nichols Basement that we feel is appropriate for solving these problems. We are working on the costs associated with this change. In the remainder of this section we comment on the specific needs which could be met by using this space.

In Fairchild we currently occupy two classrooms (F202 and F208 with approx. 1700 sq. ft.), several areas for GTA instructional staff (F203A, 203B, 203C, 203D, and 203E with approx. 1500 sq. ft.), and several class-labs (F212, F117, F14, and F5 with approx. 1600 sq. ft.). We need space in Nichols basement to consolidate all of the CIS program in one building. In addition, we need more faculty offices in Nichols for expected new faculty, thus displacing current GTA instructional staff on the third floor of Nichols. Finally, we need two new class-labs to support recitation and experimental labs for several beginning classes in programming and computing systems. One small lab of 400 sq. ft. and one large lab of 1120 sq. ft. are needed. The total space needed is thus approximately 6400 sq. ft. With space allocated for walls and hallways, this is approximately the amount space available for allocation in the basement of Nichols.

The basement space, currently inhabited by Farrell Library for storage of seldom referenced books, would accomodate all of these requirements. Thus, we could have two large bay areas for GTA staff, four class-labs, two classrooms, and some storage space that we desperately need. These components are all labeled in the attached floor plan.



J. The Plan: Summary and Time Line

In summary, in order to meet the Regents' challenge, we require eight additional faculty members ($600,000 per year) to increase the size of the graduate student enrollment, increase extramural funding, and accommodate the expected growth in the undergraduate program. This increase in faculty would also permit us to teach the introductory computing courses that are mandated by the Strategic Planning Charge to the College of Arts and Sciences; it also would permit a potential payback of two to four million dollars in extramural funding. Our requirement is for 25 workstations ($125,000) and software ($20,000 per year). A parallel computing system is also part of the requisite computing infrastructure to attract research faculty to KSU ($500,000); this system could be shared with other researchers across the University through the RIACT proposal. Additionally, workstations and file servers are needed for faculty. These systems will cost $200,000. We need funds to maintain the computing resource infrastructure at the rate of $200,000 per year. Additional graduate stipends amount to $72,000 in incremental funding.

There is a certain urgency about allocation of new faculty positions. We wish to appoint two new faculty each year for the next four years because the High Performance Computing Initiative will be put into place over the next five years and we need our senior faculty in place early enough to get some of that money. Also, we need the computing facilities within the first two years so that we can attract the high quality faculty. We also wish to pursue accreditation of our undergraduate BS program in Computer Science within the next year. We also hope to start broadcasting MS courses across NTU within two years. **Time is of the essence.**

L. References

[Boa86] Kansas Board of Regents. Mission Statements for the State Universities of Kansas, 1986.

[Gri86] Gries, D., Miller, R., Ritchier, R., and Young, P. Imbalance Between Growth and Funding in Academic Computer Science: Two Trends Colliding. Communications of the ACM, Vol. 29, No. 9. Sept, 1986.

[Gri90] Gries, D., Marsh, D. The 1988-89 Taulbee Survey Report, Communications of the ACM, Vol. 33, No. 9. September, 1990.

[Hol90] Hollings, Senator. Chairman of the Committee on Commerce, Science, and Transportation. High Performance Computing Act of 1990. Report 101-387. Report on S. 1067. July 23, 1990. U.S. Gov. Printing Office.

[Hop89] Hopcroft, J., and Kennedy, K. Computer Science Achievements and Opportunities: Report of the NSF Committee for Computer Research. SIAM Reports on Issues in the Math. Sciences, Soc. for Industrial and Applied Math. Philadelphia, Pa, 1989.

[Hud90] Hudak, P. Personal communication to Dave Schmidt on funding opportunities in programming language research. Yale University, July, 1990.

[KSU89] Planning Charge for the College of Arts and Sciences, Sept. 1989.

[NSF89] National Science Foundation Notice No. 107, Sept. 1989.

[NRC82] National Research Council. An Assessment of Research-Doctorate Programs in the United States. National Academy Press, 1982.

[Wal89] Wallentine, V. RIACT - Research Integration of Advanced Computing Technology: Enabling Interdisciplinary Research, Department of Computing and Information Sciences, Oct. 1989.

[Won90] Wong, E. Address to the 1990 Snowbird Conference, Snowbird, Utah. July, 1990.

Table 1
Undergraduate Enrollment for Fall Semesters 1982-1990

	1982	1983	1984	1985	1986	1987	1988	1989	1990
Freshmen	192	193	148	112	77	60	84	90	80
Sophomore	126	131	96	86	66	71	54	54	62
Junior	111	134	114	103	80	71	71	60	56
Senior	103	146	198	160	134	116	85	82	67
Total	532	604	556	461	357	318	294	286	265

Table 2
Graduate Enrollment for Fall Semesters 1982-1990

	1982	1983	1984	1985	1986	1987	1988	1989	1990
Master	80	65	63	83	68	67	43	43	37
Ph.D.	13	9	11	10	21	21	21	20	21
AT&T (Part-time MS)	57	62	72	70	62	50	52	51	56

Table 3
Allocated Faculty Positions FY 83 - FY 91

1983	1984	1985	1986	1987	1988	1989	1990	1991
10.5	12.5	12.5	12.0	12.5	12.5	13.5	14.0	15.0

Table 4a
Extramural Funding FY 83 - FY 90

1983	1984	1985	1986	1987	1988	1989	1990
160,000	231,734	214,639	219,435	306,337	152,422	432,535	227,418

Table 4b
Extramural Equipment and Software Grants FY 86 - 90

1985	1986	1987	1988	1989	1990
50,000	300,000	1.3M	700,000	750,000	50,000

Table 5
Faculty Publications FY 82 - FY 90

	1982	1983	1984	1985	1986	1987	1988	1989	1990
Refereed Publications	3	3	5	6	14	22	32	37	56
Books	0	3	0	1	3	1	0	0	0
Totals	8	10	10	10	19	26	32	37	56

Table 6
Computing Facilities

Type	Quantity	Equipment
Super-Mini	1	Harris HCX-9
Super-Mini	1	DEC VAX 11/780
Super-Mini	2	AT&T 3B15
Super-Mini	1	SCS-40/CTSS
Mini	10	AT&T 3B2 400
Mini	5	AT&T 3B2 310
Workstation	8	Sun 3/60
Workstation	5	Sun 3/80
Workstation	4	Sun Sparc Station 1
Workstation	1	Sun Sparc Station 1+
Workstation	3	Xerox AI System 1186
PC-Unix	60	AT&T 7300 Unix PC
PC	20	AT&T 6310 PC
PC	10	AT&T 6300 PC
PC	15	Zenith Z-150 PC
PC	15	Apple MAC
X-Terminal	4	Visual 19" X-Terminal
X-Terminal	7	NCD 16" X-Terminal
Terminal	11	AT&T 610 CRT
Terminal	10	AT&T 4425 CRT
Terminal	8	Harris 8665 CRT
Terminal	70	Various CRT Terminals
Graphics	1	AT&T Frame Creation System
Printer	4	AT&T 478 Dot Matrix
Printer	1	AT&T 495 Laser
Printer	1	Dataproducts LZR 1260i Laser
Printer	2	Centronics LineWriter 800
Printer	1	DataProducts B600
Printer	3	Okidata 84 Dot Matrix
Printer	5	Epson Dot Matrix
Printer	3	Apple ImageWriter
Printer	2	Apple LaserWriter
Printer	1	QMS LaserWriter 800
Printer	1	Xerox LaserWriter 4045
Printer	3	NEC Spinwriter 5510
Data Switch	1	Equinox Data Switch
Telephone	1	Merlin Phone System
Modem	20	Modem
Fax	1	AT&T FAX 3500D
Projector	1	Sony Projection System
Projector	1	Kodak Overhead Projection System
Network	1	Ethernet
Network	2	Appletalk
Network	1	StarLAN

Table 7
Graduate Degrees FY 82 - FY 90

	1982	1983	1984	1985	1986	1987	1988	1989	1990
Master	25	36	25	35	39	40	42	32	25
Ph.D.	0	2	2	1	1	3	2	4	3
Totals	25	38	27	36	40	43	44	36	28

Table 8
Undergraduate Degrees FY 82 - FY 90

1982	1983	1984	1985	1986	1987	1988	1989	1990
47	61	62	102	104	86	69	35	39

Table 9
Department Salaries Compared to National Average

	1985-1986	1986-1987	1988-1989	1989-1990	%Deficit
KSU Assistant Professor	36,705	37,024	41,184	46,030	
National Average	39,544	41,945	43,959	49,013	6.5%
KSU Associate Professor	36,696	37,266	42,966	48,441	
National Average	45,062	47,425	50,806	55,749	15%
KSU Professor	43,245	44,478	49,533	58,134	
National Average	59,503	63,037	69,326	72,792	25%

Table 10
OOE Funding FY 83 - 91

1983	1984	1985	1986	1987	1988	1989	1990	1991
37,336	39,236	41,590	43,669	37,119*	43,669	43,669	44,669	44,669

* 15% budget cut

Table 11
Total Student Credit Hours
FY 85 - FY 90

1985	1986	1987	1988	1989	1990
14,466	14,044	12,903	12,323	11,808	12,669

Table 12

Number of Students Enrolled in Service Courses (100 and 200 level) FY 85 - FY 90

1985	1986	1987	1988	1989	1990
3,105	2,983	2,837	2,577	2,757	2,988

Table 13
Department Scholarships

Name	Class	Fund	Amount
Teresa Detter	SO	IBM/Dean Match	1,000
Jared Friesen	JU	Phillips/Dean Match	1,000
Greg Haynes	SO	Conoco	1,000
William Smeed	SO	DST Systems	500
Robert Swenson	SO	DST Systems	500
Michael Augustine	FR	IBM/Dean Match	400
Matthew Jones	FR	IBM/Dean Match	400
Chris Luedders	FR	IBM/Dean Match	400
Rick Un	FR	Phillips Minority	1000

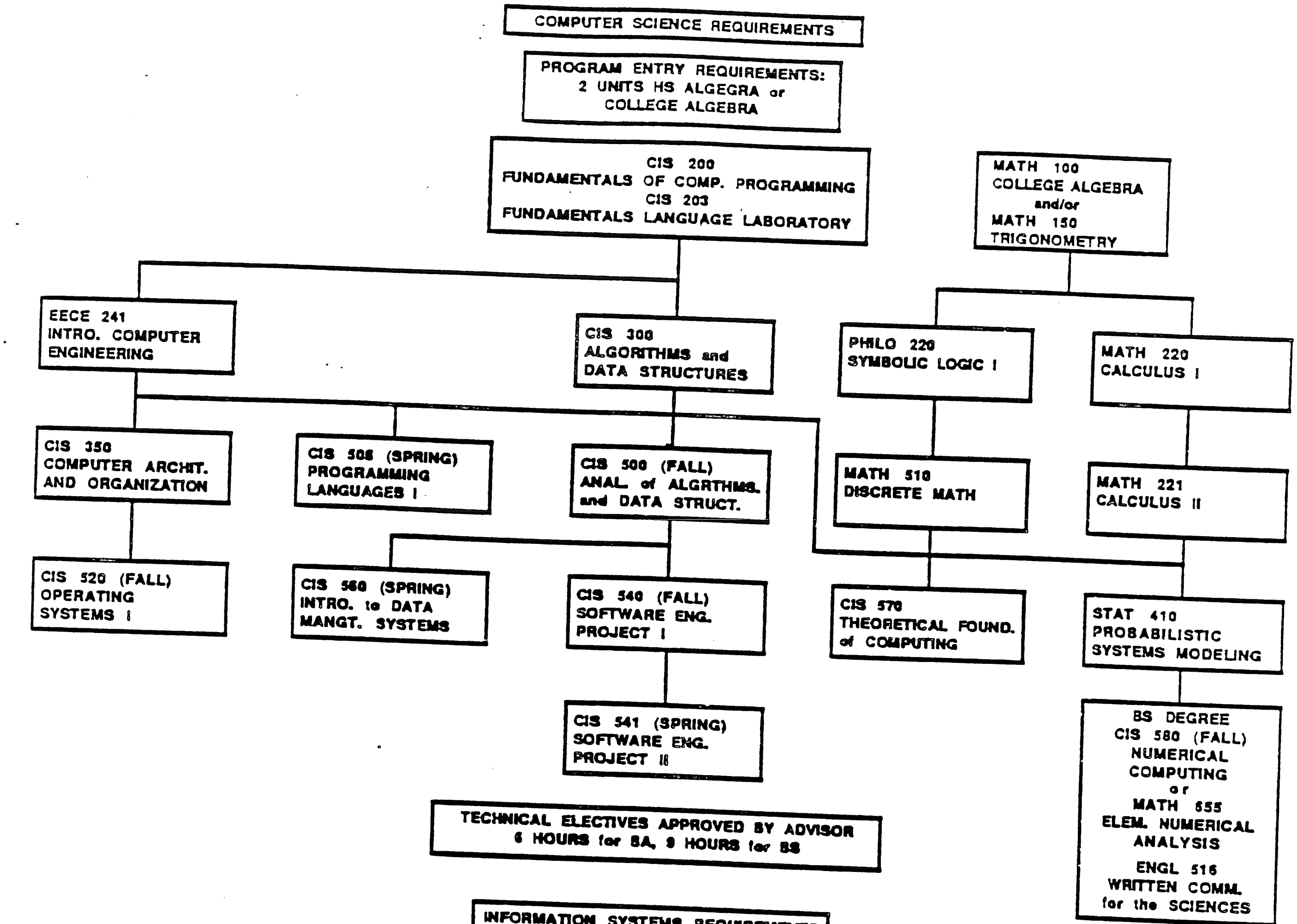
KANSAS STATE UNIVERSITY

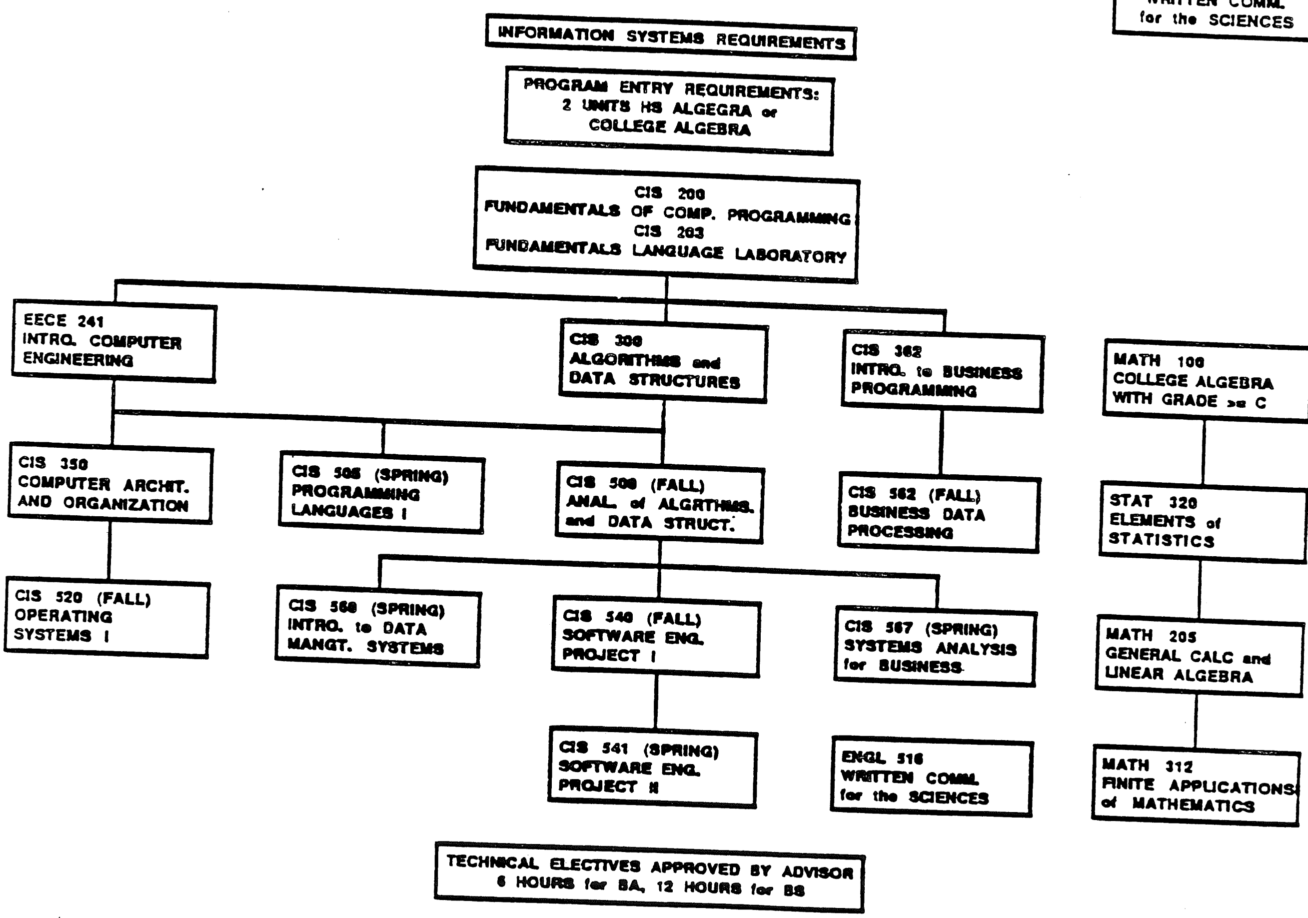
1991 GUIDE TO REQUIREMENTS

FOR

MAJORS IN COMPUTER SCIENCE & INFORMATION SYSTEMS

To major in computer science or information systems you must meet the general requirements of the University, the requirements of the College of Arts and Sciences, and the requirements of the Department of Computing and Information Sciences (all of which are listed in the General Catalog). The requirements for the BS and BA degrees are outlined on the sample curriculum guide check sheets. An up-to-date copy of the curriculum guide should be kept in your folder in the CIS office for your use during advising. Please update your guide form when you pick up your enrollment permit and take the updated version with you when you see your advisor. Please return it to the CIS office - Nichols Hall 234 - after you have been advised.


COMPUTER SCIENCE REQUIREMENTS
PROGRAM ENTRY REQUIREMENTS: 2 UNITS HS ALGEGRA or COLLEGE ALGEBRA
CIS 200 FUNDAMENTALS OF COMP. PROGRAMMING CIS 203 FUNDAMENTALS LANGUAGE LABORATORY
MATH 100 COLLEGE ALGEBRA and/or MATH 150 TRIGONOMETRY
EECE 241 INTRO. COMPUTER ENGINEERING
CIS 300 ALGORITHMS and DATA STRUCTURES
PHILO 220 SYMBOLIC LOGIC I
MATH 220 CALCULUS I
CIS 350 COMPUTER ARCHIT. AND ORGANIZATION
CIS 505 (SPRING) PROGRAMMING LANGUAGES I
CIS 500 (FALL) ANAL. of ALGRTHMS. and DATA STRUCT.
MATH 510 DISCRETE MATH
MATH 221 CALCULUS II
CIS 520 (FALL) OPERATING SYSTEMS I
CIS 560 (SPRING) INTRO. to DATA MANGT. SYSTEMS
CIS 540 (FALL) SOFTWARE ENG. PROJECT I
CIS 570 THEORETICAL FOUND. of COMPUTING
STAT 410 PROBABILISTIC SYSTEMS MODELING
CIS 541 (SPRING) SOFTWARE ENG. PROJECT II
BS DEGREE CIS 580 (FALL) NUMERICAL COMPUTING or MATH 655 ELEM. NUMERICAL ANALYSIS ENGL 516 WRITTEN COMM. for the SCIENCES
TECHNICAL ELECTIVES APPROVED BY ADVISOR 6 HOURS for BA, 9 HOURS for BS


INFORMATION SYSTEMS REQUIREMENTS
PROGRAM ENTRY REQUIREMENTS: 2 UNITS HS ALGEGRA or COLLEGE ALGEBRA
CIS 200 FUNDAMENTALS OF COMP. PROGRAMMING CIS 203 FUNDAMENTALS LANGUAGE LABORATORY
EECE 241 INTRO. COMPUTER ENGINEERING
CIS 300 ALGORITHMS and DATA STRUCTURES
CIS 362 INTRO. to BUSINESS PROGRAMMING
MATH 100 COLLEGE ALGEBRA WITH GRADE >= C
CIS 350 COMPUTER ARCHIT. AND ORGANIZATION
CIS 505 (SPRING) PROGRAMMING LANGUAGES I
CIS 500 (FALL) ANAL. of ALGRTHMS. and DATA STRUCT.
CIS 562 (FALL) BUSINESS DATA PROCESSING
STAT 320 ELEMENTS of STATISTICS
CIS 520 (FALL) OPERATING SYSTEMS I
CIS 560 (SPRING) INTRO. to DATA MANGT. SYSTEMS
CIS 540 (FALL) SOFTWARE ENG. PROJECT I
CIS 567 (SPRING) SYSTEMS ANALYSIS for BUSINESS
MATH 205 GENERAL CALC and LINEAR ALGEBRA
CIS 541 (SPRING) SOFTWARE ENG. PROJECT II
ENGL 516 WRITTEN COMM. for the SCIENCES
MATH 312 FINITE APPLICATIONS of MATHEMATICS
TECHNICAL ELECTIVES APPROVED BY ADVISOR 6 HOURS for BA, 12 HOURS for BS

KANSAS STATE UNIVERSITY
COLLEGE OF ARTS AND SCIENCES
LIST OF COURSES THAT FULFILL DEGREE REQUIREMENTS
As of March, 1990

ENGLISH COMPOSITION I & II
PUBLIC SPEAKING (or Argumentation & Debate)
PRINCIPLES OF PHYSICAL FITNESS

HUMANITIES: 4 courses, 11 hrs. minimum. One course from each of the 4 areas. They may be used at the same time to count toward the major. No course may be used to satisfy more than one specific requirement in this section. Only courses taken for two or more credit hours satisfy these requirements.

1. FINE ARTS: 1 course
   - Anthropology--Creativity & Culture 515, Afro-American Music & Culture 517
   - Art technique courses 200-799, art history 195 & 196, Intro to Museum Studies 305, Computer Imaging Art 400
   - Dance courses 205, 323, 324, 325, 326, or 371
   - History of Dance--HIST 459
   - *Music--200, 201, 245, 250, 255, 280, 310, 385, 420, 424, 455, 480, 570, 601, 602, or 650
   - Theatre courses 260-799

2. PHILOSOPHY: 1 course
   - Except Logic courses--110, 220, and 510

3. WESTERN HERITAGE: 1 course
   - *History courses in Greco-Roman, Western European or North American experience
   - Constitutional Law Courses--POLSC 613, 614, 615, 616, 799
   - Women's Studies--DAS 105, 405, 506
   - American Ethnic Studies--DAS 160
   - Political Thought Courses--POLSC 301, 661, 663,667, 671, 675, or SOCIO 709
   - Humanities Courses--ENGL 230, 231, 233, 234
   - Modern Language Courses-FREN 514, GRMN 530 SPAN 565, SPAN 566
   - Music - Intro to American Music 245
   - Speech - Rhetoric of the Sixties 460

4. LITERARY OR RHETORICAL ARTS: 1 course
   - *English courses in literature or creative writing 250-799 except 301, 400, 401, 405, 415, 490, 492, 499, 520, 516, 530, 796
   - *Modern Language literature courses, including literature in translation
   - Theatre courses 562, 764
   - Speech 330, 335, 430, 432, 434, 460, 725, 730, 732, or 733

BS Degree only: Levels I & II in the same foreign language will satisfy Western Heritage and the Literary and Rhetorical Arts requirements.

*Courses listed on following pages.

SOCIAL SCIENCES: 4 courses from 3 disciplines, 12 hrs. minimum.

Up to 2 courses from a single dept. may be used to fulfill the distribution requirements set forth in this section. They may be used at the same time to count toward the major. **One course must be 500-799 level or carry a prereq in same dept.**

At least 3 of the 4 courses must be from: Psychology, Sociology, Cultural Anthropology, (including Archaeology), Geography (except Environmental I & II (220 & 221), Economics, Political Science, History

The 4th course can be from one of the above or from the following:

- Women's Studies--DAS 105, 405, or 506
- Gerontology--DAS 315, 415
- Speech--323, 435, 520, 720, 726; Linguistics except Gen. Phonetics 601
- Journalism & Mass Communications--Intro to Mass Comm 235, Ethnic Media 530, Women & Media 612, Hist Journalism 660, Law Mass Comm. 665, Mass Comm: Ethics & Issues 685
- Radio-Television--Radio-TV & Society 300, Hist Telecomm. 660 or RTV Crit. 675
- Physical Education--Motor Dev & Learn 320, Soc. Dimen. 340, or Sport & Contemp Society 435
- Anthropology -- Pro-Seminar in Appl. Anth 640

NATURAL SCIENCES:

BS Degree: 4 courses/14 hr. min.
BA Degree--3 courses/11 hr. min.

Courses that fulfill this requirement may be used at the same time to count toward the major. No courses may be used to satisfy more than 1 specific requirement in this section. Only courses taken for 2 or more credit hours satisfy these requirements & courses in excess of 5 cr. hr. count as 2 courses.

1. A Life Science with Lab
2. A Physical Science with Lab
3. A Life or Physical Science
   - Life Sciences: Biology, Biochem., Paleobiology (Geol) 581 or 704, Intro. Physical Anthro. 280, 281, Paleoanthro. 688, Primatology 691, Osteology 694, Osteology Lab 695
   - Physical Sciences: Physics, Chemistry, Envir. Geog I & II (220 & 221) ONLY; Geol except Paleobiology 581, Paleoecology 704
4. BS Degree only: 1 course (3 cr. hr. min.) with a prereq. in the same dept chosen from the following: Life or Phys. Sci. listed in #3, Biochem courses with chem. prereq, PE--330 Kinesiology, Physio. of Exercise 335, Psych.-Psychobiol. 470, Fund. of Percep. & Sensation 480

QUANTITATIVE & ABSTRACT FORMAL REASONING.:

BS DEGREE ONLY

Courses used for this requirement may also satisfy any major requirement for which it qualifies. Select one of the following three options:

1. Three courses from: Math, Stat, Logic (Philo), Computer Sci (note: CIS 200 requires lab 203 and is equivalent to one required course)
2. One of the following pairs:
   - Quant Analysis in Geog.700 & Stat I level course
   - Meas & Eval in PE 710 & Stat. I level course
   - General Physics I 113 & Trig. 150
   - Methods in Social Res 520 & Stat. I level course
   - Intermed. Quant Meth 725 & Stat. I level course
   - Meth in Social Work Res 519 & Stat I level course
3. Level II: 2 courses

   Computer Science--Fund of Comp. Prog. 200 & lab 203 (to count as one course)

   Math--Plane Trig. 150, Applied Math 201, General Calc. & Lin. Algebra 205

   Philosophy--Symbolic Logic II 510

   Statistics--Elem. of Statistics 320, Elem. Statistics for the Social Sciences 330, Biometrics I 340, Business & Econ. Stat. I 350, Statistical Methods for Social Sci 702, Statistical Methods for Nat. Sciences 703

--OR--

Level III: 1 course

Computer Science--Algorithmic & Data Struct. 300, Comp. Architecture and Organ. 350

Math--Technical Calc I 210, Anal Geometry & Calc I 220

Philosophy--Topics in Metalogic 701

Statistics--Biometrics II 341, Business & Econ. Stat II 351, Anal. of Variance & Covariance 704, Regression & Correlation Analysis 705

BA DEGREE ONLY

Foreign Language: 4 courses 15 hrs. One of the foreign language sequences offered by the Dept. of Modern Languages or equivalent competency.

Mathematics: 1 course 3 hours

100-799 level course offered by the Dept. of Mathematics, or any other course for which there is a mathematical prerequisite. Any course used to satisfy this requirement cannot be used to satisfy any other general education requirement.

INTERNATIONAL OVERLAY:

This course may also satisfy a requirement in the major, social sciences, or humanities. The 4th course in a single foreign language sequence (other than Latin) will satisfy this requirement.

Anthropology--Intro. Cultural 200, Intro. to Ling. Anthro. 220, Intro to aeology 260, Civ. of South Asia I 505, Civ. of South Asia II 506, Folk Cultures 507, Male & Female 508, Cultural Ecology & Econ. 511, Political Anthropology 512, Creativity & Culture 515, Ethnomusicology 516, Afro-Amer. Music & Cult. 517, African American Cultures 536, Cultures of India & Pakistan 545, Cultures of Africa 550, Culture & Personality 604, Religion in Culture 618, Indians of No. Amer. 630, Indian Cultures of So. Amer. 634, Pro-Seminar in Appl. Anth 640, Precolumbian Civ. of Mexico & Guatamala 673, Archaeology of the Old World 676

Economics--Civ. of So. Asia I 505, Civ. of So. Asia II 506, Capitalism & Socialism 636, Intern'l Trade 681, Underdeveloped Countries 682

Geography--World Reg 100, Human Geog 200, Honors 201, Civ. of So. Asia I 505, Civ. of So. Asia II 506, Latin Amer. 620, Europe 640, Soviet Union 650, Geog of Hunger 710, World Population Patterns 715

History--Russian Culture & Civ. 250, Gandhi & Indian Revol. 350, Hist. of Hinduism 504, Civ. of So. Asia I 505, Civ. of So. Asia II 506, World War II 514, U.S. & World Affairs 1776-Present 543, U.S. & Soviet Relations since 1917 544, War in 20th Cent. 545, Colonial Hispanic Amer. 561, Mod. Mexico 562, Russian Revol. & Soviet Sys. 564, European Diplomatic Hist. to Napoleon 576, European Diplomatic Hist. since Napoleon 577, Russia to 1801 591, Grandeur & Decline of Imperial Russia 592, Topics in Non-Western Hist. 598

Journalism & Mass Comm--Intern'l Comm 670

Management--Intern'l Business (Bus. Adm.) 690

Marketing--Intern'l Marketing (Bus. Adm.) 544

Modern Languages--Russian Culture & Civ. 250, Russn Lit. in Translation: 19th Cent. 504, Russn Lit Trans: Soviet Period 508, Survey Russian Lit. 552

Political Science--World Politics 333, Civ. of So. Asia I 505, Civ. of So. Asia II 506, Contemp. Chinese Pol. 511, International Relations 541, American Foreign Policy 543, Pol. of Dev. Nations 545, Latin Amer. Pol. 622, So. Asian Pol. 623, Mid. East Pol. 624, SE Asian Pol. 625, African Pol. 626, Soviet-Style Regimes 627, Compar. Security Estab. 628, Admin. in Dev. Nations 629, Intern'l Conflict 642, Intern'l Pol. Eur. 645, Intern'l Law 647, Intern'l Defense Strag. 649, Intern'l Organ. 651, Intern'l Pol. So. Asia 652, Intern'l Pol. Mid East 653

Sociology--Civ of So Asia I 505, Civ of So Asia II 506, Religion in Culture 618, Soc. & Change So. Asia 742

NAME______________________________ MAJOR______________________

ADDRESS___________________________ DEGREE_____________________

__________________________________ DATE_______________________

Courses for Computer Science

Anal Geom & Calc I	MATH220	4
Anal Geom & Calc II	MATH221	4
Discrete Math	MATH510	3
Symbolic Logic I	PHIL220	3
Prob System Modeling	STAT410	3
Theo. Found of Comp.	CIS570	3

BS Degree Only

Written Comm for Sc	E516	3
Elem Numerical Anal*	M655	3
or		
Numerical Computing*	CIS580	3

Courses for Information Systems

Elem of Statistics	STAT320	3
Intro Business Prog	CIS362	3
Gen Calc & Lin Alg	MATH205	3
Finite Applications#	MATH312	3
Business Data Prog*	CIS562	3
Systems Analysis#	CIS567	3
Written Comm for Sc	ENGL516	3

Courses required for BOTH Majors

Fund. of Computer Programming	CIS 200	3
Fund. Language Laboratory	CIS 203	1
Intro. to Computer Engineering	EECE 241	3
Algor. & Data Structures	CIS 300	3
Computer Archit. & Prog.	CIS 350	3
Anal of Algorithms & Data Struct*	CIS 500	3
Intro. to Programming Languages#	CIS 505	3
Operating Systems I*	CIS 520	3
Software Engineering Project I*	CIS 540	3
Software Engineering Project II#	CIS 541	3
Intro to Data Management Systems#	CIS 560	3

* Fall ONLY
\# Spring ONLY

Technical Electives To Be Approved By Advisor:
(6 hrs for BA degree, 9-12 hours for BS degree)

1. ______________________ ____
2. ______________________ ____
3. ______________________ ____
4. ______________________ ____

Courses for Requirements for Both Degrees

See Current Listing for Courses That Fulfill Requirements

English I	3
English II	3
Oral Communications	2-3
Concepts of PE	1

Courses for BA Degree

Humanities (4 Courses) 12
1. Fine Arts ______________________ ____
2. Philosophy ______________________ ____
3. Western Heritage ______________________ ____
4. Literary or Rhetorical Arts ______________________ ____

Social Sciences (4 Courses) 12
1.______________________ ____
2.______________________ ____
3.______________________ ____
4. Courses must be 500-799 or have prereq. in same dept. ______________________ ____

Natural Sciences (3 Courses) 11
1. Life Science w/Lab ______________________ ____
2. Physical Science w/Lab ______________________ ____
3. Life or Physical Science ______________________ ____

Foreign Languages (4 Courses) 15
1.______________________ ____
2.______________________ ____
3.______________________ ____
4.______________________ ____

Math (1 Course) 3
1.______________________ ____

Courses for BS Degree

Humanities (4 Courses) 11
1. Fine Arts ______________________ ____
2. Philosophy ______________________ ____
3. Western Heritage ______________________ ____
4. Literary or Rhetorical Arts ______________________ ____

Social Sciences (4 Courses) 12
1.______________________ ____
2.______________________ ____
3.______________________ ____
4. Course must be 500-799 or have prereq. in same dept. ______________________ ____

Natural Sciences (4 Courses) 14
1. Life Science w/Lab ______________________ ____
2. Physical Science w/Lab ______________________ ____
3. Life or Physical Science ______________________ ____
4. Course w/ prereq. in same dept. ______________________ ____

Quantitative requirement is met by majoring in CMPSC or INSYS

Internat'l Overlay (1 course) 3
1.______________________ ____

AREAS OF TECHNICAL ELECTIVES 1991

COMPUTER SCIENCE MAJORS

BA select 6 hours, BS select 9 hours as follows:

Courses taken to meet the Computer Science major may not be used as technical electives. Technical electives must be Computing and Information Sciences 300 level and above. One course must be from the CIS 600 or CIS 700 levels.

INFORMATION SYSTEM MAJORS

BA select 6 hours, BS select 12 hours from the suggested coursework for a particular track.

DATABASE MANAGER

CIS 600	Microcomputer Software
CIS 761	Data Base Management Systems
MANGT 420	Management Concepts
MANGT 421	Production/Operations Management
MANGT 466	Management Information Systems

INFORMATION SYSTEMS ANALYST/DESIGNER

CIS 740	Software Engineering
ACCT 211	Financial Accounting
FINAN 450	Business Finance
MANGT 420	Management Concepts
MANGT 466	Management Information Systems
MKTG 400	Marketing

MANAGEMENT INFORMATION SYSTEMS

ACCT 211	Financial Accounting
MKTG 400	Marketing
FINAN 450	Business Finance
MANGT 466	Management Information Systems
CIS 762	Office Automation
PSYCH 560	Industrial Psychology

APPLICATIONS PROGRAMMER

CIS 600	Microcomputer Software
CIS 535	Introduction to Computer-Based Knowledge Systems
CIS 636	Computer Graphics
CIS 740	Software Engineering
CIS 745	Software Development Management

COMMUNICATIONS ANALYST

CIS 600	Microcomputer Software
CIS 750	Advanced Computer Architecture
CIS 762	Office Automation
CIS 725	Computer Networks
PSYCH 425	Problem Solving and Decision Making

Appendix 2.

GUIDELINES

FOR THE

MASTER OF SCIENCE DEGREE

IN THE

DEPARTMENT OF COMPUTING AND

INFORMATION SCIENCES

KANSAS STATE UNIVERSITY

JANUARY 1989

GRADUATE STUDIES COMMITTEE

Dr. William J. Hankley— Chair
Dr. David A. Schmidt
Dr. Virgil Wallentine

I. INTRODUCTION

These guidelines describe departmental and university requirements for a Master of Science (M.S.) Degree in Computing and Information Sciences. Students are expected to adhere to these standards. If exceptions are warranted, the student must consult the Graduate Studies Committee to determine alternate means of meeting the standards.

The guidelines stated here are those of the Computing and Information Sciences Department. Certain other regulations are imposed by the Kansas State University Graduate School and are described in the "Student Guide for Masters and Doctoral Degrees," available from the Graduate School Office, and in the "Graduate Student Handbook," published by the Graduate Student Council. It is the student's responsibility to know and satisfy all requirements.

The Graduate Studies Committee will keep each student informed of the committee's view of his or her progress towards the M.S. degree. In keeping with this commitment, an annual review of all graduate students is performed each January, and a written evaluation is transmitted to each student.

Graduate students are expected to participate in the professional activities of the Department. This includes attending seminars and colloquia, suggesting improvements in curriculum (both graduate and undergraduate), and suggesting new teaching techniques.

II. ADMISSION

The "Directions for Applying for Graduate Studies in Computer Science" manual gives detailed information about the application process. A student well prepared for graduate study will have a good background in "mainstream computer science." This includes exprience with block structured programming languages (e.g., Pascal), "modular" languages (e.g., Modula, Ada, or Smalltalk), and non-procedural languages (e.g., Lisp, Prolog, or ML), and background in computer architecture or assembly programming, data structures, operating systems, database systems, software engineering, and computing-related mathematics (e.g., mathematical logic, discrete mathematics, or calculus). A student who lacks experience in some of these areas may be asked to do specific coursework to resolve the deficiencies.

III. REQUIREMENTS FOR THE MASTER OF SCIENCE DEGREE

The M.S. degree requires a minimum of 30 credit hours of graduate level coursework; a limited number of credit hours from other accredited graduate programs can be applied. (Note: a student who chooses the "non-thesis-report" Program Option must take 33 credit hours; see Section IIIb.) Each new student is assigned a faculty member to serve as an *Academic Advisor*. The Academic Advisor helps the student select courses and reviews the student's progress until a Major Professor is selected. The coursework must include:

Background Requirement: CMPSC700; this requirement is waived if the student has already taken a course on compiler construction.

Seminar Requirement: CMPSC897. This course is an introduction to the department, general literature in computer science, and technical writing. It must be taken during a student's first year of graduate studies. Based upon the instructor's evaluation of a student's writing skills, the student may be required to take ENGL516.

Implementation Requirement: One of the courses: CMPSC620, 630, 636, 690, or 700. These courses require the student to complete a substantive software project, including specification, design, testing, and documentation.

Theory Requirement: One of the courses: CMPSC675 or 770. These courses cover formal proof techniques.

Breadth Requirement: Three of the courses: CMPSC671 (specification and verification), 705 (programming languages), 730 (artificial intelligence), 720 (operating systems), 740 (software engineering), or 761 (database systems). (Note: CMPSC762 may be substituted for CMPSC761, and CMPSC725 may be substituted for CMPSC720.) These courses give the student exposure to a breadth of areas in computing. Other courses numbered CMPSC7xx may be used to satisfy this requirement, provided that permission is granted by the Graduate Studies Committee.

Specialization Requirement: One course numbered CMPSC8xx or CMPSC9xx (excluding seminar, projects, and M.S. research courses).

The student must receive a grade of "B" or better for each course used to satisfy the above requirements.

IIIa. Advisor and Supervisory Committee

By the end of the first year as a graduate student, a student must select a *Major Professor*. The Major Professor helps the student choose a *Supervisory Committee*, pick a *Program Option*, and formulate a *Program of Study*. The Supervisory Committee is a group of three faculty members (including the Major Professor) that approves the student's Program of Study and Program Option and gives final approval for the student's degree. The final approval is granted at the *Oral Examination*, which is held when all other requirements are met for the degree. The Oral Examination is described in Section IIIc. The Program Option is described in Section IIIb. The Program of Study lists the courses that the student takes to satisfy the coursework requirements for the M.S. degree. A student must obtain a *Program of Study Form* from the Graduate School, list the courses on it, have the Supervisory Committee sign it, and return it to the Graduate School. The Program of Study Form should be completed at the end of the student's first year of studies.

IIIb. The Program Option

The Program Option can take one of three forms:

Non-thesis-report Option: Write a major paper, for example, as part of a CMPSC8xx course. This option requires 33 credit hours for a M.S. degree.

Report Option: Undertake a project that culminates in a written report; 2 credit hours for CMPSC898 are awarded for the work. Project work satisfying the Implementation Requirement can be used, subject to the approval of the Major Professor. This option requires 30 credit hours for a M.S. degree.

Thesis Option: Perform original research that culminates in a written thesis; 6 credit hours for CMPSC899 can be awarded for the work. This option requires 30 credit hours for a M.S. degree.

The document written to satisfy the Program Option should represent the best possible writing by the student; it is not to be written or extensively edited by the Major Professor. Students should begin their writing early enough so there will be time for review by the Major Professor and rewriting by the student prior to the Oral Examination. Once the student has completed the document, the student must visit the Graduate School and obtain the Graduate School's Approval Form. The Supervisory Committee members sign the Approval Form, and the student returns it to the Graduate School.

If a student chooses either the thesis or report options, the thesis or report must meet the Graduate School's standards. Tentative copies of the thesis or report are due in the Major Professor's office approximately two months prior to graduation. The Graduate School requires three copies of the thesis or report, which are submitted after the Oral Examination.

IIIc. The Oral Examination

Once the Supervisory Committee members have signed the Graduate School's Approval Form, the student returns the form and tells the Graduate School the time, date, and place of the Oral Examination. The examination should take place approximately one month before graduation, and it must occur no sooner than one week after the Approval Form is returned.

The Oral Examination is a presentation of the student's Program Option work and a defense of the student's scholarly effort. The exact format of the Oral Examination is decided by the Supervisory Committee, and the student must consult the Major Professor prior to the examination to establish the format.

A student can either pass or fail the Oral Examination, subject to a vote by the Supervisory Committee. If the student fails, a second attempt of the Oral Examination cannot be retaken in less than two weeks nor more than twelve months after the failed examination, unless an extension is granted by the Dean of the Graduate School. No third try is allowed.

IV. NORMAL PROGRESS

Each semester of enrollment, a student must make ***normal progress*** towards the M.S. degree. Normal progress is considered to be the following:

- a grade point average that is 3.00 or better.
- a Major Professor selected and a Program of Study filed with the Graduate School by the end of the first year in attendance.
- a coursework load of at least 9 credit hours per semester.

V. UNRESOLVED ISSUES

Any issues not covered in this document shall be resolved by the Graduate Studies Committee in consultation with the faculty of the Department of Computing and Information Sciences.

Appendix 3.

GUIDELINES

FOR THE

DOCTOR OF PHILOSOPHY DEGREE

IN THE

DEPARTMENT OF COMPUTING AND

INFORMATION SCIENCES

KANSAS STATE UNIVERSITY

January 1989

GRADUATE STUDIES COMMITTEE

Dr. William J. Hankley— Chair
Dr. David A. Schmidt
Dr. Virgil Wallentine

1. INTRODUCTION

1a. These guidelines give departmental and university standards for attainment of a Doctor of Philosophy (PhD) Degree in Computing and Information Sciences. You are expected to adhere to these standards. If exceptions are warranted, your advisor and the Graduate Studies Committee must be consulted to determine alternate means of meeting the standards.

1b. The guidelines stated herein are those of the faculty of the Computing and Information Sciences Department at Kansas State University. Certain other regulations are imposed by the Kansas State University Graduate School and are described in the "Student Guide for Masters and Doctoral Degrees," which is available from the Graduate School Office, and in the "Graduate Student Handbook," published by the Graduate Student Council. It is your responsibility to know and satisfy all requirements. 1c. The PhD program in Computing and Information Sciences is offered jointly by Kansas State University in Manhattan and the University of Kansas in Lawrence. Acceptance into the program implies acceptance by both departments. You may elect to fulfill residency and other requirements at either school. You may select courses from offerings at both schools.

2. ADMISSION

2a. In the usual case, you must first complete a Master's degree in computer science or a related field. See the booklets "Directions for Applying for Graduate Study in Computing and Information Sciences" and "Guidelines for the Master of Science Degree in Computing and Information Sciences at Kansas State University" for details.

2b. You must take the Graduate Record Exam (GRE). Successful applicants have a combined GRE Verbal plus Quantitative score of at least 1100.

2c. If you are an international student and you received received your Bachelor's degree abroad, you must take the Test of English as a Foreign Language (TOEFL) and achieve a score of at least 575. (The exam is waived in certain cases, e.g., for an applicant from Canada or Britain.)

2d. You must possess a grade point average of 3.50 (on a scale where an "A" is 4.00) for your Master's level coursework. Your Master's degree work must include material called the *Core Courses*. The Core Courses are:

- a compiler construction course
- a course in theoretical computer science (formal language theory or analysis of algorithms)
- three courses in "breadth areas" (artificial intelligence, database systems, operating systems, programming languages, software engineering, or systems specification)

If your Master's degree studies do not include this material, you may still apply. The Graduate Studies Commmittee may choose to admit you with the requirement that you take specific course material to remedy deficiencies.

3. GENERAL REQUIREMENTS

3a. The PhD degree requires at least 90 semester hours of graduate-level credit, typically distributed as 30 hours of Master's work, 30 hours of PhD level coursework and 30 hours of PhD research, culminating in a PhD dissertation. (See Section 4.1 below.) All work must be completed within seven years. At least one full year must be spent in residence at Kansas State University.

3b. You must maintain a 3.00 grade point average in all coursework.

3c. You must make regular progress toward completion of the degree. Progress of graduate students is reviewed each year in January by the Graduate Studies Committee. A written evaluation is sent to you and placed in your permanent file. Any student who does not maintain an adequate grade point average or who does not respond to a warning of inadequate progress will be placed on probation, with written notice from the Graduate School. A student on probation must correct deficiencies within the time limit indicated in the written notice or be dismissed from the graduate program.

3d. If you are employed by the department, you must enroll in at least 9 hours of graduate-level courses each Fall and Spring term of employment.

3e. Sometime in your graduate career you must participate in teaching within the Department, either as an assigned instructor or by special arrangement.

3f. You are expected to participate in the professional activities of the Department. You must attend seminars and colloquia offered by the Department and by the professional societies within the Departments.

4. SPECIFIC REQUIREMENTS FOR THE PhD DEGREE

4a. Upon admission to the PhD program, you are assigned an *academic advisor*, who remains your supervisor until you obtain your *major professor* (see Section 4d.). You and your advisor complete a *Declaration of Intent* form and give it to the departmental secretary. You must also consult with your advisor to formulate an initial research paper (see Paragraph 4b), a plan of study and an agreement with a research advisor (see Paragraph 4d).

4b. During your first year in the program, your academic advisor will assign you an *initial research paper* on some topic of the advisor's choosing. The paper you write must display sound organization, clear exposition, evidence of background research, and conceptual understanding of the topic. The paper does not need to be a research proposal or a new research result. The paper might relate to or be supported by a course you are taking. It should represent from 1 to 3 credit hours of work. (In some cases, you can receive CMPSC999 credit for your work.) The paper must not be edited or organized by any member of the faculty.

4c. The initial research paper will be evaluated by your academic advisor in consultation with the Graduate Studies Committee. *You will not be allowed to proceed to the second year of your PhD studies if your initial research paper is not accepted by your academic advisor and the Graduate Studies Committee.*

4d. At the end of your first year of PhD studies, you should seek a *research advisor*, also known as your *major professor*. Your research advisor must be a member of the Graduate Faculty. (See the "Kansas State University General Catalog" for further information.) Since the research advisor organizes and directs your research, you should choose an advisor carefully. How do you find an advisor? Talk to faculty members. Take some of the 800-level or 900-level research-oriented courses. Read current survey and research papers in computer science journals and magazines. *It is your responsibility to obtain a research advisor.*

4e. In consultation with your research advisor, you must compose a *supervisory committee*. The supervisory committee must include three members of the Graduate Faculty in the Computing and Information Sciences Department. Another member must be from the graduate faculty of the Computer Science Department at the University of Kansas. Another member must be a Kansas State University Graduate Faculty member from a department other than Computing and Information Sciences. All committee members must be chosen for their appropriateness to your planned research topic. In addition, the Graduate School will appoint an examination chairperson from outside of the Computing and Information Sciences Department.

4f. You must consult regularly with your research advisor.

4.1. The Program of Study

4.1a. You must meet with the members of your supervisory committe and formulate a *Program of Study*. (Obtain the Program of Study forms from the Graduate School.)

4.1b. The Program of Study contains the following information:

- 4.1b.i. major professor (that is, the research advisor)
- 4.1b.ii. members of the supervisory committee
- 4.1b.iii. general area of research
- 4.1b.iv. three preliminary examination areas (See Paragraph 4.2c.)
- 4.1b.v. all graduate course credits (at least 90 hours)

4.1c. The graduate course credits must include the following:

4.1c.i. The Core Courses stated in Paragraph 2d. Equivalent courses taken at another institution are acceptable. The Graduate Studies Committee reserves the right to determine equivalency. Alternatively, Core Courses can be omitted if you elect to take and pass the comprehensive exam. (See Paragraph 4.2b.)

4.1c.ii. At least 24 hours of course credit at Kansas State University beyond the Master's degree.

4.1c.iii. At least 30 hours of PhD research.

4.1c.iv. At least 9 hours of CMPSC900-level courses.

4.1c.v. One or more courses in theoretical or foundational topics that support your chosen direction of research. The supervisory committee approves the choice of courses for this requirement.

4.1c.vi. Any additional requirements instituted by your supervisory committee. (An example: English 516, "Written Communication for Scientists," is sometimes required for additional writing experience.)

4.2. The Preliminary Exams

4.2a. You must also pass preliminary exams. The exams consist of 4 written exams and one oral exam. *By the end of your second year of studies, you must have passed the preliminary exams.*

4.2b. The first preliminary exam is a *comprehensive exam* over the Core Courses (see Paragraph 2d). *This exam is waived if you complete the Core Courses (either at Kansas State University or at your previous school) with at least a "B" in each course and with a grade point average of 3.50 or greater for all of the courses.* There is no reading list for the comprehensive exam. The exam covers the content of of the core courses. Syllabi for the Core Courses are available from the department's Graduate Studies secretary. By the end of your second year of studies, you must have passed the comprehensive exam or satisfied the core course requirements.

4.2c. You must pass one exam from each of the following three areas:

Software Systems:
- Compilers & Interpreters,
- Distributed Systems,
- Operating Systems,
- Software Engineering

Knowledge and Information Systems:
- Artificial Intelligence,
- Data Base Systems,
- Office Automation

Theory:
- Analysis of Algorithms,
- Automata & Computability,
- Programming Language Semantics,
- Specification & Verification

4.2d. The exam areas are defined by reading lists. (See paragraph 4.2e.) You must prepare for the topics specified in the reading lists. The general scope of each area will align with a primary graduate course in each area; however, the reading lists will include some items that go beyond the primary graduate course.

4.2e. Preliminary examinations can be scheduled for either September or January. The reading lists will be available from the Graduate Studies Secretary the preceding April 1st or October 1st. You must make a written request to the Graduate Studies Committee by April 15th or October 15th to schedule your exams for the next September or January.

4.2f. The Graduate Studies Committee specifies the exam formats. Usually, the preliminary exams are 4 hours each, scheduled for 3 successive Saturday mornings. (The comprehensive exam is a five hour, closed-book exam.)

4.2g. The preliminary exams are graded by the respective faculty members who prepared them. An exam may be graded as "pass," "fail," or "conditional pass" subject to further work. If exactly one of the three exams is graded "fail," you must retake and pass that exam the next time that exams are offered. If two or more of the exams are graded "fail," you must retake and pass exams in the same three areas the next time that exams are offered. You are allowed only one retake of an exam. If you fail an exam twice, you must leave the PhD program. (The comprehensive exam can be retaken only once. If the comprehensive exam is failed twice, you must leave the PhD program.)

4.2h. The final phase of the preliminary exams is the oral exam. The oral exam occurs about a month after your written exams are graded. The format of the oral exam is set by your supervisory committee. The oral exam might cover questions of general knowledge in computer science, specific questions from your written exams, or topics in your field of research. The result of your oral exam is decided by the supervisory committee, who can vote "pass" or "fail." The committee may also decide that you must retake the oral exam a second time. You must pass the oral exam by the second try, or you must leave the PhD program.

4.2i. The Graduate School must be informed of the outcome of the preliminary exams. When you have completed two-thirds of your PhD coursework and have taken (or will soon take) your preliminary exams, ask the Graduate School to issue the ballot for the preliminary exams. The Graduate School will send the ballot to the Department, which then reports the results to the Graduate School. Upon passing the preliminary examinations you are admitted to *candidacy for the PhD degree*.

4.3. The Dissertation Research

4.3a. Once you pass the preliminary exams, you must write a *research proposal* of your dissertation research. Your proposal must present background concepts and literature, it should define the topic and goal of your research, and it should identify how you will evaluate successful completion of the goal. You must meet with your supervisory committee and present your proposal. The committee must approve your proposal.

4.3b. You must work closely with your advisor on your research, and you must write a *dissertation*.

4.3c. You must successfully defend your dissertation, subject to the following conditions:

4.3c.i. You must have been a candidate for the PhD degree for at least seven months.

4.3c.ii. You must obtain a dissertation approval form from the Graduate School. You must give each member of the supervisory committee, including the appointed Chairperson of your final examination, a copy of your dissertation and have each member sign the form.

4.3c.iii. You must allow the committee at least two weeks to read your dissertation prior to your final examination. (See Paragraph 4.3c.iv.)

4.3c.iv. You must schedule your oral presentation and defense of your dissertation (also called the *final examination*) with the Graduate School. (After you give the Graduate School the signed dissertation approval form, they will issue a ballot to the Chairperson of your final exam.)

4.3c.v. You must arrange with the Department secretary to to schedule a room and to make public announcement of the time, place, and title of your presentation.

4.3c.vi. You must present the dissertion to your supervisory committee in an open seminar, and the committee votes to "pass" the dissertation. If the committee votes to "fail," then you are allowed one retake of the defense.

4.3c.vii. If you pass the defense, you must submit the required dissertation copies, fees, and address information to the Graduate School.

4.3d. Finally, you must submit for publication at least one paper based upon your research. You must present the paper to the Computing and Information Sciences Department in an open seminar.

5. Unresolved Issues

5a. Any issues not covered in this document will be resolved by the Graduate Studies Committtee and the Computing and Information Sciences Faculty.

Appendix 4
Teaching Assignments

Calendar Year 1990 Faculty and Graduate Teaching Assistant Assignments

I. Faculty Assignments and GTA Graders

A. Professor, Associate Professor, and Assistant Professor

	Teaching Assignment Spring 1990	Fall 1990	Graduate Teaching Assistant
Virg Wallentine	CMPSC 690	CIS 690 CIS 990	Jim Butler
Bill Hankley	CMPSC 505	CIS 636 CMPSC 840	Jim Peters (spring)
Elizabeth Unger	Sabbatical	CIS 960	
Myron Calhoun	CMPSC 305 CMPSC 362 CMPSC 500	CIS 362	Peikun Tsai Azfar Moazzam (spring) Anindya Banerjee (spring)
David Gustafson	CMPSC 541 CMPSC 740	CIS 535 CIS 540	Richard Courtney (spring) Jim Peters (fall)
Austin Melton	CMPSC 370 CMPSC 990	CIS 606 CIS 990	Kasinath Vemulapalli (spring) Dennis Ng (fall)
Dave Schmidt	CMPSC 806 CMPSC 990	CIS 705 CIS 990	
Maarten vanSwaay	CMPSC 490 CMPSC 492 CMPSC 520	CIS 350 CIS 500	Sudhukar Ramakrishna (spring) Azfar Moazzam (fall) Anindya Banerjee (fall)
Maria Bleyberg	CMPSC 730 CMPSC 830	CIS 630 CIS 890 CMPSC 890	Muralidhar Venkatrao (fall)
Jan Chomicki		CIS 761	Dennis Ng
Olivier Danvy		CIS 570	Adrain Fiech
Rodney Howell	CMPSC 675	CIS 870 CMPSC 990	Mitch Neilsen (spring)
Masaaki Mizuno	CMPSC 620 CMPSC 720 CMPSC 798	CIS 520 CIS 720	Mitch Neilsen

K. Ravindran	CMPSC 725	CIS 825	Ka Wing Wong (spring)

B. Instructor and Instructor-Temp.

	Teaching Assignment Spring 1990	Fall 1990	Graduate Teaching Assistant
Joseph Campbell	CMPSC 567 CMPSC 897	CIS 562 CIS 897	
Charles Kichler	CMPSC 110		
Clark Sexton	CMPSC 200 CMPSC 207	CIS 300	Eric Fong (spring) Mohammad Paryavi
Kole Scarbrough			Mini Supercomputer Administration

II. GTA Assigned as Classroom Teachers

Troy Anderson	CIS 204 (fall)
Ed Coburn	CIS 110 (fall)
Cindy Cook	CIS 203 (fall)
Amit Halder	CMPSC 110 (spring)
Kiang Pang	CMPSC 110 (spring)
Jim Slack	CMPSC 200
Charles Black	CMPSC 206 (spring); CIS 208 (fal)
Glen Diener	CIS 110 (fall)
Steve Hansen	CMPSC 560 (spring)
Abdul Kasim	CMPSC 206 (spring); CIS 203 (fall)
Sheela Ramanna	CMPSC 207 (spring)
Kevin Lynn	CMPSC 211 (spring)
Mohammad Paryavi	CMPSC 300 (spring)
Tom Talkington	CIS 110 (fall)

III. Miscellaneous GTA Assignments

Thenmozhi Arunan (fall,grd 110)
Vivek Bansal (fall,grd 203)
Jeff Brogden (systems)
Adrian Fiech (spring,grd 300)
Puneet Gupta (fall,grd 110)
Janaki Krishnaswamy (grd 110)
Dennis Ng (spring,grd 560)
Sudhukar Ramakrishna (fall,grd 203)
Raghavendra Rao (spring,grd 110)
Manhohan Sankhla (grd 110)
Kasinath Vemulapalli (fall,grd 203)
Ka Wing Wong (spring, grd 110)

David Balda (spring, coordinate 200)
Baba Prasad (fall,grd 110)
Kyung Doh (spring,grd 20X)
Eric Fong (fall,grd 200)
JR Hockersmith (systems)
David Liu (fall,grd 20X)
Peter Prakash (spring,systems;fall grd 203)
Sheela Ramanna (fall,grd 208)
S. Samdarshi (spring,grd 110;fall,grd 204)
M. Nelakonda (grd 110)
M. Venkatrao (spring,grd 110)

Appendix 5a
Departmental Committees
Department of Computing and Information Sciences 1989-90

1. **Faculty Recruiting**
   This committee will have a tremendous impact on the future of the department because the competition for faculty is very high. This committee will develop strategies for recruiting faculty. All faculty will participate in trying to recruit specific candidates.
   Melton, Mizuno, Wallentine (Chair), and Zamfir

2. **Undergraduate Studies Committees - Hankley (Chair)**
   The responsibilities of this committee are to develop curricula for undergraduate majors, coordinate with the college curriculum committee, coordinate with the graduate studies committee, and make recommendations on entrance and continuation requirements. This committee is also charged with developing service courses for majors in other departments. Thus it must coordinate with other departments on campus to provide up-to-date courses which prepare all college students to work in an information-intensive workplace.
   a. *Computer Science and Information Systems Majors Subcommittee*
      Gustafson (Assoc. Chair), Howell, Sexton, and Van Swaay.
   b. *Service Courses Subcommittee*
      Calhoun, Campbell (Asoc. Chair), Kichler, and Slack

3. **Graduate Studies**
   This committee must monitor the graduate curriculum, screen applicants for grad. school, coordinate with the University of Kansas on the PhD program, recruit graduate students, and coordinate with the UG studies committee.
   Hankley (Chair), Schmidt, and Wallentine

4. **Seminar Series**
   This committee is responsible for coordinating speakers for a seminar series within the department. This includes recruiting local faculty and graduate students (including KU), regional faculty, ACM lecturers, faculty candidates, and 2 national speakers each year.
   Gustafson (Chair), Howell, and Melton

5. **Computing Facilities**
   This committee must make recommendations on the acquisition and modification of computer hardware and software tools. This includes tools for the mainframe, minis, and micros. This committee will also formulate policy on the use of the departmental computing facilities. It must also coordinate with all faculty and staff to acquire teaching and research tools.
   Harris, Mizuno, Townsend, and Wallentine (Chair)

6. **Faculty Evaluation Review Committee**
   The task for this committee is to review procedures for reappointment, tenure, and merit salary increase, and make recommendations to the faculty and department head. The resulting procedures, upon ratification by the faculty, will be applied by the department head.
   Schmidt (Chair), Gustafson, and Wallentine

Appendix 5b
Committee Service

Maria Zamfir-Bleyberg

Faculty Recruiting
Faculty Evaluation Review Committees.

Myron Calhoun

Undergraduate Studies Committee

Jan Chomicki

Graduate Studies Committee

David Gustafson

Faculty Recruiting Committee
Scholarship Review Committee
Undergraduate Studies
Departmental Seminar Committee
Faculty advisor to ACM Student Chapter

William Hankley

Graduate Studies Committee
Undergraduate Studies Committee

Rod Howell Undergraduate Studies Committee
Seminar Series Committee

Austin Melton

Faculty Search Committee
Graduate Advisory Committee
Faculty Evaluation Committee
CCOP

Masaaki Mizuno

Faculty Recruiting Committee
Faculty Evaluation Committee

K. Ravindran

Computing Facilities Committee

David Schmidt

Faculty Evaluation Committee
Graduate Studies Committe

Elizabeth Unger

Undergraduate Studies
Physical Sciences subcommittee of the Graduate Council until Aug 15
Chair: Dean's Advisory Committee in Arts and Sciences until Aug 15
Strategic Planning subcommittee on Graduate Education

Maarten van Swaay

Undergraduate Studies Committee

Virgil Wallentine

Faculty Recruiting Committee
Computing Facilities Committee
Faculty Evaluation Committee

Appendix 6
Faculty Publications

Published or Accepted

Baker, A., Bieman, J., Fenton, N., Gustafson, D., Melton, A., and R. Whittey. (1990). "A Philosophy for Software Measurement", J. Systems Software 12:3, pp277-281.

Balda, D., and D. Gustafson. (1990). "Cost Estimation Models for the Reuse and Prototype Software Development Life-Cycles", SEN 15:(3), pp 42-50.

Baruah, S., Howell, R., and Rosier, L. "Algorithms and Complexity Concerning the Preemptive Scheduling of Periodic, Real-Time Tasks on One Processor." To appear in Real-Time Systems.

Baruah, S., Howell, R., and Rosier, L. (1990). "On Preemptive Scheduling of Periodic, Real-Time Tasks on One Processor." 15th International Symposium on Mathematical Foundations of Computer Science, LNCS 452, pp. 173-179, Banska Bystrica, Czechoslovakia.

Bleyberg, M., Isenhour, T., Marshal, J., and T. Zhou. "The Design and Implementation of an Analytical Chemistry Expert System" presented at IEA/AIE-90 (the 3rd International Conference on Industrial and Engineering Applications of AI and Expert Systems), Charleston, SC, July 1990.

Butler, J., and V. Wallentine. (1990). Vignettes: A Distributed Discrete Event Simulation System. International Phoenix Conference on Computers and Communications, Phoenix, AZ.

Butler, J., and V. Wallentine. (1990). Message Bundling in Time Warp. SCS Multiconference on Advances in Parallel and Distributed Simulation, Anaheim, CA.

Cabrera, M. and E. Unger. (1990). "Dynamic Data as Deterrent to the Tracker," Proceedings or the 1990 Sigsmall/PC Symposium on Small Systems.

Chomicki, J., and V.S. Subrahmanian. (1990). "Generalized Closed World Assumption is Pi-0-2 complete" Information Processing Letters 34, 289-291.

Chomicki, J. (1990). "Polynomial-Time Query Processing in Temporal Deductive Databases" Proc. Ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Nashville, Tennessee.

Courtney, R., and D. Gustafson. (1990). "Evolution of Architectural Design Representations", Proceedings of the 1990 Symposium on Applied Computing.

Danvy, O., and A. Bondorf. (1991). ''Automatic Autoprojection of Recursive Equations with Global Variables and Abstract Data Types'', to appear in Science of Computer Programming.

Danvy, O., Jones, N., Gomard, C., Bondorf, A., and T. Mogensen. (1990). ''A Partial Evaluator for the Lambda-Calculus'', Proceedings of the IEEE Computer Society 1990 International Conference on Computer Languages, pp. 49-58.

Danvy, O., and C. Consel. (1990). ''From Interpreting to Compiling Binding Times'', Proceedings of the European Symposium on Programming ESOP 90, pp. 88-105.

Danvy, O., and A. Filinski. (1990). ''Abstracting Control'', Proceedings of the 1990 ACM Conference on Lisp and Functional Programming, pp. 151-160.

Danvy, O., and C. Consel. (1991). ''Static and Dynamic Semantics Processing'', Proceedings of the 1991 ACM Conference on Principles of Programming Languages.

Danvy, O., Friedman, J., and J. Franco. (1991). ''The Scheme Programming Language'', in ''A comparative study of parallel programming languages: the Salishan problems'' (John Feo, Ed.), Elsevier Science Publishers, The Netherlands.

Even, S., and D. Schmidt. (1990). Type inference for action semantics. In Proc. 1990 European Symp. on Programming, Lecture Notes in Computer Science 432, Springer-Verlag, Berlin, pp. 118-133.

Fenton, N., and A. Melton. (1990). Deriving structurally based software measures, The Journal of Systems and Software, Vol. 12, pp. 177-187.

Gouda, M., Howell, R. and Rosier, L. "The Instability of Self-Stabilization." Acta Informatica 27 (1990), pp. 697-724.

Gustafson D., Melton, A., An, K., and I. Lin. (1990). "Software Maintenance Models" in IEEE Tutorial on Software Maintenance and Computers, (D. Longstreet, ed.), pp 23-35, IEEE Computer Society Press.

Hagemann, C. and E. Unger. Fuzzy Sets in Multi-level Decision making (A LAN Small Group DSS), Cybernetics and Systems: An International Journal.

Hankley, W., and J. Peters, "Temporal Specification of Ada Tasks", Jan 90 Hawaii Conf on System Sciences.

Hankley, W., and J. Peters. (1990). "Proving Specifications of Tasking Systems Using Ada/TL", Tri-Ada 90 Conference, Baltimore, MD.

Hankley, W., and J. Peters. (1991). "A Proof Method for Ada/TL Specifications", 8th Conf on Ada Technology, Atlanta, GA.

Hankley, W. (1991). Tutorial on "Formal Specification of Programs" accepted for SIGCSE Conference, San Antonio, TX.

Howell, R., Rosier, L., and Yen, H. "A Taxonomy of Fairness and Temporal Logic Problems for Petri Nets." To appear in Theoretical Computer Science, special issue for MFCS '88.

Howell, R., Rosier, L., and Yen, H. "Global and Local Views of State Fairness." To appear in Theoretical Computer Science.

Keller-McNulty, S., McNulty, M., and D. Gustafson. (1990). "Stochastic Models for Software Science". Accepted for publication in Journal of Systems and Software.

Liu, Y.D., Wong, K.W. and E.A. Unger. (1990). "Using Active Messages to Implement Office Procedures," ACM 1990 Symposium on Applied Computing.

Main, M., Melton, A., Mislove, M., and D. Schmidt, editors. Proc. 5th Conf. on Math. Foundations of Programming Language Semantics, Lecture Notes in Computer Science 442, Springer, Berlin, 1990.

McNulty, S. and E. A. Unger, "Database System Security: Inferential Attack'" National Academy of Science, Invited Paper 1991.

Melton, A., Gustafson, D., Bieman, J., and A. Baker. (1990). "A Mathematical Perspective for Software Measures Research", Software Engineering Journal 5:5, pp 246-254.

Mizuno, M., and M. Neilsen. (1991). Decentralized Consensus Protocols. 1991 Phoenix International Conference on Computers and Communications.

Ng, Y., Melton, A., and E. A. Unger. (1990). "A Method for Constructing Generalized Non-Normal Form Models'" Proceedings of the 19th Annual Computer Science Conference, ACM.

Perng, J. and E. Unger. (1990). "A User Friendly Front-End for MPS," ACM 1990 Symposium on Applied Computing.

Peters, J. F., Ramanna, S., and E.A. Unger. (1990). "Logic of Knowledge and Belief in the Design of an Integrity Kernel for an Office Information System,' ACM Annual Computer Science Conference(poster session).

Peters, J. F., Ramanna, S., and E. A. Unger. (1990). Design of Knowledge-based Integrity Systems with ISL++". International Conference on Software Engineering and Knowledge Engineering.

Ramanna, S., Peters, J.F., and E. A. Unger. (1990). "Temporal Specification of an Integrity Kernel for Multimedia Ofice Systems," ACM 1990 Symposium on Applied Computing.

Ramanna, S., Peters, J.F., and E.A. Unger. (1990). "Nonmonotonic Logic in the Specification of an Integrity System". Proceedings of 13th National Computer Science Conference.

Ramanna, S., Peters, J.F., and E.A. Unger. (1990). "Logic of Knowledge and Belief in the Design of Distributed Integrity Kernels," PARBASE-90: International Conference on Database, Parallel Architectures, and their Applications.

Ramanna, S. Peters, J.F., and E.A. Unger. (1990). "Design of Knowledge-Based Integrity Systems with ISL++," International Conference on Software Engineering and Knowledge Engineering.

Ranft, I., and D. Gustafson. (1990). "Using the Software Process Model to Analyze a Software Project", Proceedings of CompSac90.

Ravindran, K. (1990). Application-specific Group Communications in Distributed Servers. 10th International Conference on Computer Communications, IEEE INFOCOM 91, Miami, FL.

Ravindran, K. (1990). A Flexible Broadcast Communication Interface for Distributed Applications. Submitted to 11th International Conference on Distributed Computing Systems, IEEE ICDCS.

Ravindran, K. (1990). Protocol Bypass Concept for High Speed OSI Data Transfer. 2nd International Workshop on Protocols for High Speed Networks, IFIP WG6.1/WG6.4, Palo Alto (CA).

Saiedian, H. and E. A. Unger. (1990). "ABSL: An Actor Based Specification Language for Office Automation," ACM Computer Science Conference.

Saiedian, H., and E. Unger. (1990). " A Formal Specification Tool for Distributed Office Systems". Proceedings of the 1990 Sigsmall/PC Symposium on Small Systems.

Schmidt, D. (1990). Action semantics-based language design. Invited paper, Proc. SOFSEM90 winter school, Janske Lazne, Czechoslovakia, 30 pp.

Shenoi, S., and A. Melton. (1990). An extended version of the fuzzy relational database model, Information Sciences, Vol. 52, pp. 35 - 52.

Shenoi, S., Melton, A., and L.T. Fan. (1990). An equivalence classes model of fuzzy relational databases, Fuzzy Sets and Systems, Vol. 32, pp. 153 - 170.

Shenoi, S. and A. Melton. (1990). A set-relational interpretation of similarity and proximity relations. 4th International Workshop of the Bellman Continuum, KSU.

Unger, E. A. and S. Keller-McNulty. (1990). "The Deterrent Value of Natural Change in a Statistical Database," Proceedings of 13th National Computer Security Conference, Washington D.C.

Unger, E., Harn, L., and V. Kumar. (1990). "Measures of Database Information based upon Information Entropy". Proceedings of 6th Computer Security Applications Conference.

Unger, E., S. McNulty, and P. Connelly. (1990). "Natural Change in Dynamic Databases as a Deterrent to Compromise by Trackers". Proceedings of the 6th Computer Security Applications Conference.

Wong, K. W., and E. A. Unger. (1990). "Specification and Verification of Active Message Systems,". Proceedings of the 1990 Sigsmall/PC Symposium on Small Systems.

Submissions

Bleyberg, M., Isenhour, T., Marshal, J., and T. Zhou. "On Concurrency Control in an Analytical Chemistry Expert System" submitted to the International Journal of Applied Intelligence.

Bleyberg, M. "An Entity-Relationship Algebra" submitted to the International Journal of Foundations of Computer Science.

Bleyberg, M. "Modeling Concurrency with AND/OR Algebraic Theories" submitted to 2nd International Conference on Algebraic Methodology and Software Technology, Iowa City, Iowa, May 1991.

Chomicki, J., and T. Imielinski. "Finite Representation of Infinite Query Answers" 45 pages. Submitted to ACM Transactions on Database Systems.

Chomicki, J. "Depth-Bounded Bottom-Up Evaluation of Logic Programs" 30 pages. Submission in Journal of Logic Programming.

Chomicki, J. "Efficient Maintenance of Dynamic Integrity Constraints Using Materialized Temporal Views" 17 pages. Submitted to 1991 ACM SIGMOD Conference.

Dybkjaer, H., and A. Melton. (1990). Data types and dialgebras. Submitted to the Proceedings of Workshop on Category Theory Applied: Paradigmatic Topics, Bremen, Germany.

Dybkjaer, H., and A. Melton. Comparing Hagino's categorical Melton, A., and S. Shenoi. Fuzzy relations and fuzzy relational databases. Submitted to International Journal of Computers and Mathematics with Applications.

Dybkjaer, H., and A. Melton. Comparing Hagino's categorical programming language and typed lambda-calculi. Submitted to Theoretical Computer Science.

Even, S., and D. Schmidt. Category-sorted algebra-based action semantics, Journal of Theoretical Computer Science, accepted for publication and in press.

Even, S., and D. Schmidt. Type inference for action semantics. In Proc. 1990 European Symp. on Programming, Lecture Notes in Computer Science 432, Springer-Verlag, Berlin, 1990, pp. 118-133.

Hagemann, C. and E. A. Unger. (1990). Theoretical Foundations for a Multilevel SGDSS. Submitted to MIS Quarterly.

Hansen, S., and E. A. Unger. (1990). "An Extended Memoryless Inference Control Model". Submitted to Sigmod 1991.

Hansen, S., and E. Unger. (1990). "Horizontal Partitioning a tool for inference control". Submitted to Applied Coomputing Conference.

Hsieh, S., Unger, E., and R. Mata. (1990). Static Analysis of Programs for Observable Information Flow. Submitted to Journal of Systems and Software.

Hsieh S., Unger, E., and R. McBride. (1990). Dynamic Slicing Algorithms. Submitted to IEEE TOSE.

Hsieh, S., and E. Unger. (1990). Obserable Data Flow for Security Analysis. Submitted to Journal of Systems and Software.

Howell, R., Rosier, L., and Yen, H. "Normal and Sinkless Petri Nets." Submitted to Journal of Computer and System Sciences.

Liu, D., and E. A. Unger. (1990). "An Integrity Model based on Knowledge." Submitted to Applied Computing Conference.

Main, M., Melton, A., Mislove, M., and D. Schmidt, editors. Proc. 5th Conf. on Math. Foundations of Programming Language Semantics, Lecture Notes in Computer Science 442, Springer, Berlin, 1990.

Melton, A., Ng, Y., and E. Unger. (1990). "Minimality fo Normalized Complex Objects on Plotkin Powerdomian Ordering". Submitted to PODS 1991.

Melton, A., Schroeder, B., and G. Strecker. Lagois connections. Submitted to SIAM Journal of Computing.

Melton, A., Ng, Y., and E. Unger. (1991). Transforming normalized relations into complex object views. Submitted to International Conference on Mathematical Foundations of Database and Knowledge Base Systems, Rostock, Germany.

Melton, A., Ng, Y., and E. Unger. (1991). Syntactically and semantically correct normalized complex objects. Submitted to ICALP'91, Madrid, Spain.

Melton, A., Schroeder, B., (1991). Connections. Submitted to the 7th International Conference on the Mathematical Foundations of Programming Semantics, Pittsburgh, Penn.

Melton, A., and S. Shenoi. (1991). Partition relational database model: an extended abstract. Submitted to International Conference on Mathematical Foundations of Database and Knowledge Base Systems, Rostock, Germany.

Shenoi, S., and A. Melton. Restricted Domain Partioning: A mechanism for establishing contexts. Submitted to IEEE Transactions on Knowledge and Data Engineering.

Mizuno, M., Neilsen, M., and K. Pang. (1990). A Generalized Token Based Distributed Algorithm for AND-Synchronization. Submitted to IEEE Transaction on Parallel and Distributed Computing.

Mizuno, M., and M. Neilsen. (1990). Generalized Assertion Based Distributed Algorithms for AND-Synchronization. Submitted to Distributed Computing.

Mizuno, M., and M. Neilsen. (1990). Read and Write Coteries. Submitted to Information Processing Letters.

Mizuno, M., and M. Neilsen. (1990). Coterie Join Algorithm. Submitted to IEEE Transaction on Parallel and Distributed Computing.

Mizuno, M., and M. Neilsen. (1990). A Dag-Based Algorithm for Distributed Mutual Exclusion. Submitted to the 11th International Conference on Distributed Computing Systems.

Mizuno, M., Halder, A., and M. Neilsen. (1990). A Node Recovery Algorithm for Assersion-based Distributed Mutual Exclusion Algorithms. Submitted to the 11th International Conference on Distributed Systems.

Mizuno, M., Neilsen, M., and R. Rao. (1990). A Token Based Distributed Mutual Exclusion Algorithm based on Bicoteries. Submitted to the 11th International Conference on Distributed Systems.

Mizuno, M., and D. Schmidt. A denotational semantics-based correctness proof of a security flow control algorithm. Journal of Formal Aspects of Computing, submitted Dec. 1990.

Ramanna, S, Peters, J.F. and E. Unger. (1990). Nonmonotonic Logic in the Specification of an Integrity System. Submitted to ACM TOIS.

Ravindran, K. (1990). Tradeoffs Between Complexity and Efficiency of Distributed Application Protocols. Submitted to IEEE Transactions on Computers.

Ravindran, K. (1990). Correcting Remote References to a Server in Distributed Operating Systems. Submitted to Computer Networks and ISDN Systems (North-Holland Publ. Co.).

Ravindran, K. (1990). A Flexible Broadcast Communication Interface for Distributed Applications. Submitted to 11th International Conference on Distributed Computing Systems, IEEE ICDCS, Oct. 1990.

Ravindran, K. (1990). A Model of Naming for Fine-grained Service Specification in Distributed Systems. Submitted to ACM Symposium on Small and Personal Computers, ACM SIGSMALL/PC, Nov. 1990.

Schmidt, D. Action semantics-based language design. Invited paper, Proc. SOFSEM90 winter school, Janske Lazne, Czechoslovakia, Nov. 1990, 30pp.

Unger, E. A., and S. McNulty. (1990). Data Compromise in a Dynamic Database. Submitted to IEEE Data and Knowledge Engineering.

Appendix 7
Grantsmanship

Funded

Maria Zamfir-Bleyberg

Second year of National Science funding for "ANALYTICAL DIRECTOR - An Artificial Intelligence/Robotic Expert System for the Analytical Laboratory", Professor T. Isenhour (principal investigator) and I (co-investigator).

Travel Faculty Development Award for $1500.

David Gustafson

NATO, "Formal Foundations of Software Measurement", #0343/88.

"A Proposal for Copperative Research in Software Measurement", NSF International Programs Sep 1, 1990.

"Developing a Formal Prosess Model for Software Re-engineering Environments" AFOSR proposal (with Eric Byrne), November 1, 1990.

Austin Melton

ONR Grant N00014-88-K-0455, through Sept. 1991.

NATO Collaborative Research Grant 034/88, through Summer 1991.

K. Ravindran

"Architectures and Protocols for High Speed Packet Switching in High Speed Multi-service networks", research initiation grant ($11,000) from Kansas Technology Corporation to support two graduate research assistants for the period Nov. 1989 to July 1990.

"Software Systems for Fault-tolerant Industrial Applications", research initiation grant ($13,764) from Kansas Technology Corporation to support two graduate research assistants for the period Aug. 1990 to May 1991.

David Schmidt

NSF Grant CCR-8822378, Semantics-directed compiler synthesis, June 89-May 91, $157,000.

NSF Grant INT-9014042, Semantics-directed compiler synthesis: travel, Jan 91-Dec 94, $12,000.

Elizabeth Unger

CRCCA Grant 91E014, Data Integrity in Data Systems, August 90 - May 91, $9,800.

CRCCA Grant 91E015, Inferential Data Security in Data Systems, August 90 - May 91, $9,800.

Virgil Wallentine

CRCCA Grant 91E012, Temporal Locality in parallel and Distributed Discrete Event Simulation, August 90 - May 91, $18,234.

Pending:

Jan Chomicki

NSF, Dynamic Integrity Constraints in Databases, $70,000.

David Gustafson "A Proposal for Copperative Research in Software Measurement", NSF International Programs Sep 1, 1990.

"Laboratory for Office Automation and Direct Manipulation" NSF equipment proposal (Bill Hankley), November 16, 1990.

William Hankley

NSF, Laboratory for Office Automation and Direct Manipulation, $17,197.

Austin Melton

NSF Research Travel Grant with Dave Gustafson

Sun Microsystems Grant for equipment.

K. Ravindran

"Design and Implementation of a Flexible Broadcast Communication Interface for Distributed Applications", grant proposal submitted to National Science Foundation for funding ($174,087) for the period from June 1991 to May 1993 to cover purchase of SUN-SPARC workstations and software, summer support for self and support of two graduate research assistants

Elizabeth Unger

NCSC, $61,326, Inferrential Theoretical Approaches to Modelling.

David Schmidt

British Science and Engg. Research Council proposal, Research in semantics and concurrency at Univ. of Edinburgh, July-Aug. 1991, approx. $2000.

NSF proposal, Action semantics and partial evaluation (with Olivier Danvy), June 1991-May 93, $190,000.

Rejected:

Jan Chomicki

Faculty Development Award, 1991 SIGMOD International Conference, $900.

BGR Award, Research in Deductive Databases, $450.

Rod Howell

NSF, (with Mizuno, M., Ravindran, K., and D. Schmidt). Programming Languages & Distributed Computing Laboratory, $1,365,691.

Austin Melton

NASA, Development of a Lattice-Theoretic Relational Database Model for Data Abstraction, $??.

K. Ravindran

Faculty Development Award, Data Driven Communication in Distributed Operating Systems, $3,000.

BGR, Architecture and Protocols for High Speed Packet Switching in Multi-Service Networks, $1,750.

NSF, A Data Driven Communications Architecture for Distributed Operating System, $70,000.

Virgil Wallentine

Hewlett Packard, Proposal for an Electronic Studio, $1,471,172.

AT&T, Proposal for Multiprocessor for Graduate Education and Research, $800,000.

Appendix 8

Current Research Programs of the CIS Faculty

Research in this department can be categorized in five basic areas - programming languages, software engineering, knowledge engineering, data base systems, and parallel and distributed systems. In this section we list the current specifc research projects of the CIS faculty.

Maria Zamfir, Ph.D., UCLA. Her research interests include different but interacting areas: the initial algebra semantics of parallel distributed computing, neural networks, and formal semantic models for the design of databases and knowledge-based systems.

In the area of parallel computing, her goal is to develop a language for writing and testing formal specifications of parallel distributed systems based on the AND/OR net model. The AND/OR net model is an initial algebra semantics model for concurrent computing systems, which I have been working at for the past few years. I have also been examining Petri nets as object-oriented systems in which abstract data types provide values for attributes. I have been using this view of Petri nets to define an abstract operational semantics for them based on "reflection". Finally, I hope that the study of neural networks will open new directions in my research in the area of parallel computing.

Regarding databases and knowledge-based systems, she is interested in building practical systems with appropriate logical foundations. At present, she is involved in the design and implementation of an expert system that can design and simulate an analytical chemistry procedure and controls the robot during the procedure execution. Regarding databases, she has been working at the implementation of an object-oriented database. This implementation is based on a formal categorical model of databases, which I have developed.

Myron A. Calhoun, Ph.D., Arizona State. Trying to delve deeply into the uses of Finite Inductive Sequences (FIS) as described by Fisher & Case. FIS appears to be directly applicable to the compression of textual data as well as compressing, processing, and recognizing visual images; this latter may also include applications in mobile free-ranging robotics. His ongoing (but now mostly background) research emphasizes the application of computers to real-world problems such as the development of computer interfaces for the handicapped and low-cost packet-radio networks."

David A. Gustafson, Ph.D., Wisconsin-Madison. His research interests are in the area of software engineering. He is formalizing the theory of software measures so that it becomes obvious what is being measured and what propoerties the measure has. He is also doing research into the problems of validating software measures. Another area of research is software reliability. He is currently investigating models of the software structure that can be used to develop a software reliability model. Related to the area of reliability is the area of software testing methods. He is developing more thorough test methods that have formal bases. Another area in which he is involved is the area of formal notations for diagrams, both data flow diagrams and hierarchy diagrams. The creation of better notations will allow more formal work on transformations of the diagrams. Finally, he is working on developing notations for describing the software development process in terms of the documents that are produced.

Rodney Howell, PhD, University of Texas at Austin. His research interests lie mainly in three areas: real-time scheduling, self-stabilization, and Petri nets. In the area of real-time scheduling, he has been looking at the complexity of finding valid schedules for various types of recurring real-time task systems. In many cases, the problems turn out to be NP-hard. His goal is to identify as many situations as possible in which schedules can be constructed efficiently. Regarding self-stabilization, he is interested in examining various theoretical limitations for self-stabilizing systems. For example, he has recently explored situations in which certain types of models cannot simulate other types of models while preserving self-stabilization. Finally, in the area of Petri nets, he has been examining the computational complexity of various problems, such as reachability, boundedness, equivalence, liveness, and fair nontermination, for different classes of Petri nets. His main goal in this area of research is to tighten the known bounds on the complexity of the reachability problem for Petri nets.

William J. Hankley, Ph.D., Ohio State University. His research centers on formal specification of programs. Writing formal specifications is a kind of programming; it is the use of very high level non-procedural languages. The research focus is on object-oriented and modular structure (using ADA concepts), high level data types (sets, maps, sequences as in VDM), logic specifications (predicate calculus and Prolog notations), and temporal description of task behaviors (temporal logic). Related work includes formal verification of specified system properties, development of executable specifications as program prototypes, and use of direct manipulation interfaces for rapid development of prototypes.

Austin Melton, Ph.D., Kansas State University. His research interests include programming semantics, software engineering, complex objects, and category theory. In programming semantics he is interested in using category theory to understand and explain programming semantics, and further he is interested in seeing how category theory itself can be used as a programming language. In software engineering he works with software measures or metrics. His work involves trying to develop a foundation upon which one can with confidence design and define useful software measures. In databases he is working to define a general method for defining and studying non-normal forms structures.

Masaaki Mizuno, Ph.D., Iowa State University. Research interests are in various aspects of distributed systems. He has worked on an information flow control mechanism for modular programming systems. He also works with Dr. David A. Schmidt on theoretical aspects of information flow by applying methodology in programming semantics. In his research in distributed systems, he and his students have developed efficient distributed mutual exclusion algorithms and distributed AND-synchronization. Currently, his group is studying concurrency control and recovery issues of transaction based distributed database systems.

K. Ravindran, Ph.D., British Columbia. Currently pursuing research on distributed systems architectures and high speed packet networks. Specific areas being investigated are: (i) Data-driven communication in distributed operating systems to allow fine-grained reconfigurability of services and fine-grained parallelism among functions that compose a server; (ii) Design of a flexible communication kernel for distributed applications whereby different applications may choose different forms of communication mechanisms to suit their requirements; (iii) Network architectures and protocols to handle congestion control, bandwidth management and packet multicasting in high speed packet switching.

David Schmidt, Ph.D., Kansas State University.Pursuing research on the theory of programming languages as it is expressed within denotational semantics. He uses denotational semantics to analyze the structure of programming languages and to implement them. In past research, he has shown how to synthesize efficient implementation data structures for languages defined by denotational semantics. He and a research student are building a "rapid prototying," compiler synthesis system based on these ideas.

Recently, he has studied the category-theoretic foundations of a denotational semantics variant called "action semantics." He and a student have developed a sound and complete type inference algorithm for action semantics; the algorithm is being implemented as part of a programming language analysis "workbench."

Elizabeth Unger, Ph.D., University of Kansas. The entire thrust of her research program is in the development of security and integrity systems based upon the object oriented programming paradigm. The work proceeds with two foci: description of the general inference problem and characterization of the database administrator and user level integrity constraints. The first thrust includes the completion and documentation of the value of natural change for deterrent value on the tracker attacks; the mathematical and statistical characterization of the security value of such change; the security value of change in conjunction with other deterrent methods; the characterization of information increment given a user data increment. This latter characterization is just beginning with Shannon's concept of entropy as the basis for measurement. Such a measure will allow the use of a semantic model to characterize statistically the security risk of releasing data in certain risk environments. The second thrust is concerned with the formal description of one aspect of user level integrity, the temporality. In this thrust, a next step is the clear definition of user level integrity, the specification of a language in which to spedify contraints (to be used in the security project also) and the definition of the architecture of such a system within contemporary operating systems.

Maarten van Swaay, Ph.D., Leiden (Netherlands). Interests in laboratory instrumentation and in neural network systems. He has written a chapter on laboratory computing for a handbook on chemical instrumentation; the book is scheduled for publication in March 1990. In addition to technical areas Dr. van Swaay has a strong interest in social and ethical issues of computing, and has developed a course in that area in our department.

Virgil Wallentine, Ph.D., Iowa State University. Research includes parallel and distributed systems and their applications. More specifically, his work centers on what can be distributed, how it can be distributed across multiple processing units, and what properties of the system make it amenable to distribution. Presently, he is working in the area of Parallel Discrete Event Simulation (PDES) and in methods for debugging distributed programs. Specific emphasis are on study of a formal language semantics for the time-space model of synchronization and a study of temporal behavior of PDES. Several specific projects are on-going which include the construction of a system which supports a visual programming facility for queueing networks, a performance prediction environment for PDES, and a knowledge-based debugging system for distributed programs.

Appendix 9
Professional Activities of the Faculty

Maria Zamfir-Bleyberg

None

Jan Chomicki

Editor: Workshop on Deductive Databases
Reviewer:
- ACM Transactions of Database Systems
- Journal for Computing & Systems Sciences
- ACM Computing Reviews
- International Conference on Database Theory

Talks:
- University of North Carolina, Chapel Hill
- University of Pittsburgh
- Arizona State University
- Texas A & M
- University of New Mexico
- Ohio State University

Olivier Danvy

Referee:
- PLILP 1990
- RTA 1991
- MFPS 1991
- PEMP 1991
- ACM Transactions on Programming Languages and Systems, 1990

Talks:
- Stanford University
- Northeastern University
- MIT
- Harvard University
- Yale University
- Carnegie Mellon University

David Gustafson

Talks:
- Topeka chapter of DPMA

William Hankley

Reviewer:
- ACM Computing Reviews
- Hawaii Systems Conference
- IEEE Transactions on Software Engineering

Rod Howell

Referee:
- IEEE Transactions on Parallel & Distributed Systems
- 11th International Conference on Distributed Computing Systems

Austin Melton

Referee:
- NSF
- IEE Software Engineering Journal

Program Committee for 7th Intl Conf on Mathematical Found of Prog Semantics
Program Committee for CSC 1992

Masaaki Mizuno

Conference Stream Chair, 1991 Symposium on Applied Computing

K. Ravindran

Referee:
- IEEE Computer Journal

Dave Schmidt

Reviewer:
- 1991 Conf on Mathematical Foundations of Programming Semantics
- Conf on Partial Evaluation & Semantics Based Programming
- Louisiana Board of Regents
- Journal of Formal Aspects of Computing
- Journal of Automated Reasoning
- John Wiley Publishers
- Oxford University Press

Elizabeth Unger

Reviewer:
- Editorial Board, Journal of Information Management Systems ACM Computing Reviews Prentice Hall Addison Wesley Little Brown Benjamin Cummings Merrill

Reviewer of conferences:
- ACM CSC ACM/IEEE WAC 1990

Consultant to the Louisiana Board of Regents on Computer Education
Leadership and Organization:
- ACM Sigsmall Vice Chair (elected office by ~7000 members) ACM Sigapp Secretary (appointed office this year) ACM Sigsamll General Chair of the 1990 Conference WAC 1991 Co-program chair for annual meetings

Talks:
- University of Missouri--Kansas City

Virgil Wallentine

Program Co-Chair for 1992 National Computer Science Conference.
Reviewer:
- Harper Collins Publishers
- IEEE Software

1990 ACM Symposium on Applied Computing
Reviewed tenure decisions for two other CS departments
Program committee for 1990 ACM Symposium on Applied Computing
Program committee for 1990 ACM Symposium on Personal and Small Computers

A Beginner's Guide to PhD Research

Computing and Information Sciences Dept.
Kansas State University
Manhattan, Kansas

November, 1989

Preface

Several years ago, I chanced upon an internal report distributed by the Artificial Intelligence Department at Edinburgh University. The report, called *The Researcher's Bible**, was a helpful list of do's and don't's for the beginning PhD student. I have distilled from that report information useful to PhD students in the Computing Sciences Department at Kansas State University and added material specific to the requirements of our Department. The Computing Sciences Faculty have proofread the result and made several improvements (I hope!). I hope that you, the beginning PhD student, will find this report helpful to you in your PhD studies.

David Schmidt
November, 1989.

---

* By Alan Bundy, Ben Du Boulay, Jim Howe, and Gordon Plotkin. D.A.I. Occasional Paper 10, Artificial Intelligence Dept., Edinburgh University, Edinburgh, Scotland, Sept. 1978.

Introduction

If you are a PhD student, your objective is to obtain a Doctor's degree. Through the actions you take to get the degree, you learn to do research. What is research? A dictionary states that it is "scientific investigation or inquiry." Such a definition is of little help when you attempt research for the first time.

This report provides some guidelines for doing research and obtaining the PhD degree.

The general requirements for the PhD degree in computing science are:

1. Coursework: You must accumulate 24 credits of coursework work beyond the M.S. degree.
2. Initial research paper: During your first year of studies, you must write a paper on a topic of your choosing.
3. Preliminary exams: By the end of your second year of studies, you must pass 3 written exams and 1 oral exam.
4. Dissertation: You must research, write, and successfully defend a dissertation.

The exact requirements for 1. to 4. are given in the "Guidelines for the PhD Degree" booklet that is distributed by the Department, so we will not repeat them here. In the sections to follow, we will describe some strategies you can undertake to satisfy the requirements and obtain your degree.

I. Coursework

Coursework gives you background for doing dissertation research. The topics of PhD-level courses are not "cut and dried"; issues of current debate in the computing research community are often presented. When attending these courses, you should be a questioning, "skeptical" student: ask questions of the instructor, the course readings, and yourself. Do the topics of the course seem important to computing? Do the proposed solutions sound reasonable? Are any important issues overlooked? Is the instructor presenting the material in a way that makes it interesting and useful to you?

Such questions awaken your budding research skills. A person can do research only if there is an interest and a need to investigate a topic. You can encourage your research intuitions by striving to work beyond the requirements of your courses. Extra background reading, completion of optional homework exercises, and discussions with classmates outside of lectures can help. Discussions are particularly helpful, because new ideas are often created in conversations with others. Also, coursework and research are more fun when people perform it in common.

The University requires that you accumulate 24 coursework credits; this takes about three terms to satisfy. You should spend the first one or two terms "shopping around" for interesting topics and instructors. By the beginning of the third term, you should make a decision as to which areas of computing interest you the most. The third (and subsequent) terms should be spent taking advanced seminars in those areas. If your chosen area of interest does not match the courses offered in your third term of study, you might contract with a faculty member to study a topic independently for course credit CIS798, CIS890, or CIS990. This is an excellent way to prepare for researching, as independent background reading is essential to any research work.

II. Academic Advisor

Your initial efforts in the PhD program are supervised by your academic advisor, who is assigned to you on your entry into the program. The academic advisor's responsibilities are to help you schedule your coursework, monitor the writing of your initial research paper, encourage you to select a Supervisor, and guarantee that you will be prepared to take your preliminary exams on schedule. (See Sections III.-V. below.)

The research interests of your academic advisor might not match your initial interests, and you should not assume that your academic advisor will become your research supervisor. Your academic advisor acts as a ''temporary supervisor'' until you find a permanent one.

III. Initial Research Paper

By the end of your first year, you are required to write a paper, called the *initial research paper*, on a topic of your choosing. Writing the paper gives you a first experience at reading background literature, collecting ideas, and presenting the ideas in a unified way. The Department uses the term paper to verify that you have elementary reading and writing skills. (It is better to discover such shortcomings the first year than to deal with them in the midst of writing a dissertation!)

You are encouraged to write your paper in combination with a course that you take in the first year of studies. Don't worry about whether or not the topic of the paper matches your ''real'' interests— the purpose of the paper is to test your writing skills, *not* to start you on the first draft of the dissertation! Obtain your academic advisor's permission before you start the paper. If you have no topic in mind for the paper, your academic advisor will suggest one.

The paper should be 10 to 20 pages in length, with a bibliography of 6 to 10 references.

IV. Supervisor

The most important step you take at the end of your first year of PhD coursework is the selection of a major professor (hereafter called your *supervisor*). Your supervisor's primary duty is to manage your dissertation research, but your supervisor also helps you form a supervisory committe, schedule exams, and handle other administrative matters.

Select a supervisor that is *capable*, *concerned*, and *compatible*. First, your supervisor must be capable of understanding and managing your research work; this normally requires that the supervisor has researched and published in the area you plan to study. Second, your supervisor must be concerned enough about the topic you select that the supervisor will take a personal interest in your work. Finally, you and your supervisor must have compatible personalities, because the two of you will be working together for several years.

In a small department like ours, it is sometimes impossible for you to obtain your first choice of supervisor. Some professors have a "full load" of advisees, and they are unable to work with any additional students. If this happens to you, don't get discouraged. Consider other professors and remember that, at this early stage of your studies, there are many research areas that will prove interesting to you once you learn a bit more about them.

With the cooperation of your supervisor, you will formulate a *program of study*, which lists your coursework, supervisory committee, and chosen area of research. Your coursework and committee must complement your chosen area of research: coursework should include specialty courses in the research area, and the committee should include people who research in your chosen area. The department head has final approval of the program of study.

V. Preliminary Exams

By the end of your second year, you take the three written preliminary exams. By passing the exams, you verify that you have fundamental knowledge in three main areas of computing. You should select the three exam areas based on topics you like and topics you might eventually research, since preparing for the exams becomes more pleasant when you know that your studying is preparing you for the first stage of your dissertation work. The exams are based on the material in the fundamental graduate courses, so you should take the fundamental courses in the three areas you select for the exams. A list of courses and topics is given in the "Guidelines for the PhD degree" booklet.

Reading lists for the written exams are available approximately 3 months prior to the exams. Contact the chairman of the graduate studies committee to obtain the reading lists. You should start studying as soon as you receive the reading lists. Some students wait too late to begin studying and thereby place unneeded pressure on themselves. You may find it helpful to form a study group with other students who are taking common exams. But beware— do not trust other students to research an exam topic and then give their notes to you! Their notes may be incomplete and inaccurate, so read for yourself all the items on the reading lists. Earlier versions of the exams are usually available. Ask the chairman of the graduate studies committee for copies.

The usual format for each exam is a closed-book, four hour, pencil-and-paper test, but you should check with the chairman of the graduate studies committee for verification. An exam will contain factual questions, questions that require analysis of standard results, and questions that require synthesis or construction of new results. The exams are meant to be a bit difficult, so don't be too distressed if you fail one or more of them the first time. You are given a second chance to pass the exams, and you will certainly learn more about an area if you take its exam twice!

Meet with your supervisor as soon as you pass the written exams to schedule the oral exam. Your supervisor will contact your supervisory committee members and set the format of the exam.

Your supervisory committee will review your academic progress at the oral exam. The committee may choose to test your knowledge of computing, although they are more likely to focus on your intended research directions. Be prepared to describe your research interests and possible dissertation topics; in turn, your committee will contribute ideas towards your research proposal. Based on your performance, the supervisory committee votes to pass or fail you. If the committee votes to pass, they will sign your preliminary exam ballot and return it to the Graduate School.

VI. Dissertation

The bulk of your time in the PhD program will be spent on research. The research leads to the writing of a dissertation, which is the evidence you submit to the Department and University that you can perform original research of high quality.

What is a dissertation? The *Graduate Faculty Handbook*, which is the University's standard reference for such matters, merely states:

> Regardless of the form used, a thesis or dissertation shall be sufficiently complete so as to allow an independent investigator or scholar to repeat and/or verify all of the work leading to the author's results and conclusions. In certain cases, where a manuscript prepared for publication is to be used, the terseness or page restrictions required by a professional journal may prevant an author from meeting this condition with the publishable manuscript alone. In such instances the thesis or dissertation must include additional materials which will insure independent reproducibility.

This information isn't very helpful! The reason the description is so vague is that the format for research and the dissertation vary from discipline to discipline. Computing, currently being a half-science, half-engineering discipline, has no set tradition for its research and dissertations.

The following quote from the article *Computing as a Discipline** sheds some light on what should be expected of a computing science dissertation:

> The three major paradigms, or cultural styles, by which we approach our work provide a context for our definition of the discipline of computing. The first paradigm, *theory*, is rooted in mathematics and consists of four steps followed in the development of a coherent, valid theory:
>
> (1) characterize objects of study (definition);
> (2) hypothesize possible relationships among them (theorem)
> (3) determine whether the relationships are true (proof);
> (4) interpret results.
>
> The second paradigm, *abstraction* (modelling), is rooted in the experimental scientific method and consists of four stages that are followed in the investigation of a phenomenon:
>
> (1) form a hypothesis;
> (2) construct a model and make a prediction;
> (3) design an experiment and collect data;
> (4) analyze results.
>
> The third paradigm, *design*, is rooted in engineering and consists of four steps followed in the construction of a system (or device) to solve a given problem:
>
> (1) state requirements;
> (2) state specifications;
> (3) design and implement the system;
> (4) test the system.
>
> Computing sits at the crossroads among the central processes of applied mathematics, science, and engineering. The three processes are of equal— and fundamental— importance in the discipline, which is a unique blend of interaction among theory, abstraction, and design.

* By Peter Denning, et. al, *Comm. ACM* 32-1 (1989) 9-23

Your PhD research should encompass at least one of the three paradigms just described. The dissertation is a documentation of the steps you took to fulfill the paradigm. A dissertation in computing might span more than one paradigm (e.g., a dissertation based on the design paradigm might also contains proofs of correctness of parts of the implementation, or a dissertation based on the theory paradigm might also contain implementation of some of the concepts studied), but the dissertation's main characteristic is that *an idea is developed and then validated by means of a rigorous proof, a simulation, or an implementation.* The validation must clearly demonstrate that the research idea is a sound contribution to computing knowledge.

How much work goes into the dissertation research? People have suggested various ''rules of thumb''; we suggest that the results in a dissertation should equal or exceed the results found in one quality journal paper. This does not mean that you must publish a journal paper before you receive the degree, but some minimum quality and quantity of results must be met before the supervisory committee deems the research completed.

VII. Research Topic

Your supervisor will recommend background readings in your area of interest. You should also visit to the Departmental and University libraries and explore the reports, books, and journals related to your area. Write a note card for each paper or book that you study (or would like to study, if you can't find enough time!). This helps remind you what you've learned and what you've liked.

Your supervisor will soon ask you to try your own hand at problem solving. You may be asked to duplicate experiments, implementations, or proofs similar to the ones found in your readings. Problems uncovered in others' research will present themselves, and these often provide starting points for your own work. Or, your supervisor might present a family of problems in your area and ask you to try solving some of them. Soon, you and your supervisor decide upon the problem area you will research.

In *The Researcher's Bible**, Bundy, et. al., state these criteria that a problem should satisfy:

1. You and your supervisor must be enthusiastic about solving it.
2. The solution of the problem must be worthy of a PhD degree, that is, the results you create must satisfy the ''one journal paper'' rule.
3. It must be within sight of the state of the art, that is, it must be solvable within three years of research work.

The importance of criteria 1. cannot be overstated! Dissertation research is hard work, and you will need all the enthusiasm you can muster. Your supervisor's enthusiasm is also important, because you must count on your supervisor's help when you are lost or stuck. In addition, not only will you be working on your research problem for several years to obtain your degree, but when you continue work after your schooling, you will likely be building on your results.

When you choose a problem, avoid the following traps:

1. Solving the world: Don't pick a research goal that's too ambitious. Read the literature and talk to fellow workers to find out what the state of the art is. One good source of ideas is

* All of the numbered lists given here onwards are adapted from *The Researcher's Bible.*

the "further work" section of research publications. Read the literature carefully. Another starting point is redoing others' bad work, but properly.

2. Manna from heaven: Don't choose a topic with no obvious starting point. It does no good to sit in your room with a blank piece of paper and a pencil, waiting for insights to come down from above.
3. Hacking: Don't just write code. You can spend years at a terminal, modifying and extending code. You get a sense of achievement when an error is exposed or a nice output is printed. This "progress" is illusionary. Your program must be explainable at a higher level than code for it to make a real contribution to computing. Recall the design paradigm: a system should be implemented only after requirements and specifications are set.

VIII. Research Proposal

Before you begin your dissertation research, you must have the approval of your supervisory committee. The approval is given at the research proposal meeting. The purpose of the meeting is to demonstrate to your committee that you have conducted adequate background reading, you have chosen a problem to investigate, and you have selected a paradigm and specific techniques to solve the problem.

You must write a proposal paper. It should contain:

1. A statement of the problem.
2. A survey of the area in which the problem arises and earlier efforts at solving the problem.
3. A description of the proposed research, stating the paradigm to be used to solve the problem, how the problem will be solved, and what are the expected results.
4. An account of any work you have done on the problem to date.
5. An approximate timetable.
6. A bibliography.

Expect your supervisory committee to ask critical questions. They must verify that you have chosen a problem worthy of a PhD degree and that you have the preparation and potential to complete the degree. Expect your committee to suggest revisions to the problem, the solution method, and the timetable. And if your proposal meets with skepticism, expect your committee to require a second proposal paper and meeting.

A change of supervisor or research topic necessitates another research proposal meeting.

IX. Research

Doing research is difficult, full-time work, and you should treat it no differently than any other full-time job: you must work regular, significant hours, and you must expect that progress will be uneven. You must also accept the following "facts of life":

1. Academic life is competitive and lonely: One fears one's own failure and the success of others. Often one's research interests are not shared by coworkers.
2. Getting down to work is hard: There are always pressures which make it easy to avoid research work, e.g., teaching or coursework. Starting in the morning is almost always hard, as is "changing state" from teaching to attending lectures to researching.

3. Solving problems is hard: One tends to avoid them and hope that they will go away or one tries to solve them all at once.
4. Reading is difficult: The difficulty seems to depend on the stage of academic development. Initially it is hard to know what to read. Later, reading becomes seductive and is used as an excuse to avoid research. Finally, one lacks the time and patience to keep up with reading (and one fears to find evidence that one's own work is second-rate or one is slipping behind).
5. The dissertation seems all-or-nothing: Once one has embarked on doing a dissertation, there is no "safety net," e.g., an "almost-PhD" diploma.

This list is a bit depressing, but it's important for you to know that everyone who does research faces the very same hurdles that you do. So, don't get discouraged if you encounter one of the above "facts" in your own work!

You can make best progress at research if you develop good work habits. Here are some tips that you might find helpful:

1. Getting started: Make a regular working schedule, and stick to it. It doesn't have to be 9am to 5pm, but there should be a definite time of day when you start work. Otherwise, you will find yourself postponing research with endless, routine, domestic chores. When you do start your work each day, begin with something easy. A good strategy is to finish a section or small problem that you left unfinished from the night before. Or, find an easy task associated with the work that you can do, e.g., a diagram or some typing. Combat the "blank sheet of paper" syndrome by getting a binder labelled "Dissertation" into which you can put bits of the research as you develop them.
2. Combatting isolation: Find a friend to whom you can talk when you need feedback. Write or send E-mail to others who work in your field. See your supervisor on a regular basis— at least once a week— and give your supervisor notes that describe what you have been doing. This creates a basis for discussion.
3. Imposing structure on your ideas: When faced with a problem, divide it into smaller subproblems. Tackle the subproblems one at a time. If a problem seems too hard to solve in its current form, and it is not clear how to subdivide it any further, try to solve a simpler version of the problem and then generalize your simpler solution to a solution for the more general case. (Example: Rather than proving all programs in a computer language have "Property X," begin by proving that some subset, say, the "structured programs," have Property X. If you can't prove it, try restricting the set of programs further or weakening the property to a "Property Y" such that Property X implies Property Y. Once you complete a proof, analyze it and try to generalize it to a larger class of programs or a stronger property.)
4. Writing: Write notes and papers to yourself. The notes serve as a "diary," documenting what approaches did and did not work. They also serve as a sourcebook for inspiration and strategies that might be useful in the future. Once you have made concrete progress on the research, draft a working paper that your friends and supervisor can read. Ultimately, these working papers will form a first draft of your dissertation.
5. Imposing a structure on your working life: Set short term deadlines for yourself. Scan the journals and newsletters for announcements of conferences in your area, and write papers by the submission deadlines for the conferences. If your paper has concrete results that seem of quality equal to those in published papers, submit the paper for possible publication.

6. Avoiding research roadblocks: Sooner or later, you will encounter a problem that you just can't solve. Rather than falling into the trap of the "blank sheet of paper," admit (temporary) defeat and back up. Go to the library and read the literature related to the problem you are trying to solve. It is amazing how much better one understands a research paper when one has been trying to solve a problem similar to the one described in the paper! Talk to your supervisor and friends. Often, new insights arise when you describe your difficulties to others. Finally, don't be afraid to back away from a problem for a month or two. Give your mind a rest and work on some other topic related to your research. It is not unusual for a researcher to spend a year trying to solve a small, well-defined problem, so don't get discouraged if you work several weeks and can't find a solution.

Your research will take one or more years, and it is possible that you will adjust your original research problem and solution strategy based on the work you do over that time. Any significant change of problem or solution method (e.g., changing from the theory paradigm to the design paradigm) must be reported to your supervisory committee.

Finally, you might wonder when you have accumulated enough results for your dissertation. Remember the "journal paper rule": the results should be of quality and quantity equal to that found in one quality journal paper. Your supervisor can help you decide whether your work has reached this point.

X. Dissertation Document

When is it time to write the dissertation? Some people begin writing once the bulk of the research is completed, and clear cut results have been achieved. Others will write while the research is being conducted. (Background chapters can be written almost any time.) If you have been fortunate and published research papers in conferences or journals, use the papers as a basis of a first draft of the dissertation.

The dissertation presents the results of your research and satisfies the promise of your proposal paper. It should include:

1. A statement of the problem.
2. A survey of the area in which the problem arises and earlier efforts at solving the problem.
3. A description of your research work, stating the paradigm used to solve the problem, the specific methods used, and the problem's solution.
4. Conclusions and future work.
5. Bibliography.

Your supervisory committee's job is to verify that your dissertation achieves the standards set by the University and Department for a degree. Don't surprise your committee by solving a different problem than the one you presented at your proposal meeting. In turn, they may surprise you! It is a good idea to meet with each of the members of your committee a month or two before your final examination so that any potential wrinkles are ironed over. Your supervisor can be of great help in ensuring a smoothly run final examination.

XI. Publications

The Department requires that you write at least one research paper based on the results in your dissertation. The paper should reflect the very best aspects of your work. It is the evidence to the world that you have made a useful contribution to computing. You are writing the paper not only to satisfy a Departmental requirement but to help your fellow researchers.

There exist a wide range of journals, conferences, and workshops that print papers on computing. Journal publications describe completed research projects, conference papers present results of work in progress, and workshop papers state preliminary results or proposals. Standards of refereeing vary from excellent to haphazard to nonexistant, but as a general rule, journals have better quality refereeing than do conferences than do workshops. Many academic departments count only journal papers as "publications" due to the lack of uniform quality refereeing in conferences and workshops.

Your goal should be the publication of your paper in the best journal in your research area.

Try to write clearly and simply. The most important results are useless if they can not be explained clearly to others. Model your paper after those papers you read that you found useful and enjoyable. Keep the paper brief and to the point.

There is no time restriction on when the paper is written: you may choose to write a research paper before, during, or after the writing of the dissertation. Your supervisor can help you decide.

Conclusion

Virtually no one is a "born researcher." The skills needed for research are best developed through study of others' results and one's own practice and persistence. The techniques listed in this booklet have been used successfully by many researchers in computing. Give them a try.