Academic Program Review of Computer Science

Department of Computer Science
University of Victoria
March 2007

Purpose of the Academic Program Review

The purpose of the Academic Program Review is to assist the department in evaluating the quality and suitability of its academic endeavours. The assessment is to facilitate both short and long-term planning.

Purpose and Scope of this Document

This document provides both historical and current material on all aspects of the Department of Computer Science and is meant as background for the review process. It addresses the four main facets of the department: undergraduate education, graduate education, research and infrastructure.

Executive Summary

In the last ten years, the Department of Computer Science has made remarkable progress. The number of faculty members has almost doubled, enabling the department to build a very strong research core in almost all areas of the discipline. From this core has come a dense web of collaborative partnerships between members within Computer Science, between Computer Science and a multitude of other departments on campus (including Humanities, Social Sciences, Human and Social Development and Fine Arts, as well as Science and Engineering), and beyond UVic (the University of Victoria), both nationally and internationally. We were the first university in Canada to introduce a set of exciting new combined programs between Computer Science and other disciplines including Visual Arts, Music, Psychology, Health Information Science and Geography.

Much of the expansion of the department in the last five years has been funded under the provincial government's *Double The Opportunities (DTO)* program, designed to increase enrolment in Computer Science, and Electrical and Computer Engineering. The start of DTO, unfortunately, coincided with the peak in enrolment, so the resulting undergraduate enrolment since then has been considerably less than the target numbers.

EXCELLENCE IN TEACHING AND LEARNING

Members of the department have attracted a very large share of the grants given by the Learning and Teaching Centre at UVic to enhance the quality of our teaching and courses with innovative ideas and pedagogical structures. This has contributed to the overall excellence of our undergraduate teaching.

Severe strains resulted from the sharp increases in enrolment in 1999 - 2002, but, in the last few years, we have shared the challenge of rapidly falling enrolment faced by all computer science departments in North America. Luckily, our numbers have not fallen as rapidly as the average, and recruitment strategies for the best students has been enthusiastically embraced and practiced by all members of the department.

A major successful initiative is to encourage the recruitment and retention of women in computer science programs, enabling us to avoid the North American trend for a reduction in the percentage of women in computer science programs.

Development of new sets of learning outcomes and investigation of new alternate paradigms for effective teaching and learning are leading to revisions and enhancements at all levels of our programs, with particular emphasis, at present, on the first two years. With one of the strongest co-op programs in Canada, experiential learning is, and has always been, a major part of our undergraduate education.

Like other computer science departments, we also play a large service teaching role for many programs across campus. We are proud of our success in effectively meeting the needs of all these groups.

RESEARCH IMPACT AND RECENT GROWTH

The research programs of the department have flourished in the last few years, and have been strengthened by the addition of outstanding new faculty members as well as three Canada Research Chairs. In the current year, the three new faculty positions in graphics have brought the result of very strong research programs in almost all areas of the discipline. With a well established base of interaction with large numbers of external groups as well as many wide ranging multidisciplinary projects, the department is ideally situated for the future.

The graduate program has grown rapidly and is closely integrated with the research endeavours. However, external grants to faculty provide almost three quarters of the funding for graduate students at all levels and this becomes an increasingly serious limitation on the ongoing expansion of the graduate program. At the present time, we have one of the two largest thesis-based graduate programs in the university.

Graduate co-op and experiential learning have become an increasingly valuable and innovative part of our graduate education.

After three years of hard work, the department moved into its exciting new building in May 2006. Our consolidation into a single location with state of the art facilities has been very beneficial. Given the recent decline in undergraduate enrolment, it is clear that the period of very rapid expansion of the last five years is now at an end, so the department can reflect on exciting new initiatives for the coming years.

THE BOTTOM LINE

Computer Science is a highly collegial department that benefits from the support of an outstanding dedicated staff. It is an exciting department still going through the very rapid changes in the discipline. We are lucky to have the people, the energy, and the vision to secure our place as one of the stronger Computer Science departments in Canada.

Table of Contents

Executive Summary i

Table of Contents v

List of Tables ix

List of Figures xi

Chapter 1: Overview 1

1.1 History of UVic and the Computer Science Department....................1
1.2 Purpose of the Department 2
1.3 Department Assessment 3

Chapter 2: Undergraduate Education 5

2.1 Overview....................5
2.2 "Traditional" CSc Programs....................6
2.3 Interdisciplinary Programs 7
2.4 Software Engineering....................9
2.5 Service Courses 10
2.6 Types of Courses and Prerequisites 11
2.7 Curriculum Renewal 14
2.8 Co-operative Education 16
2.9 Enrolment Trends in Undergraduate CSc at UVic....................18
2.10 Student Demand and Performance....................20
2.11 Teaching Quality....................28

Chapter 3: The Graduate Program 31

3.1 Structure of the Programs....................31
3.2 Courses in the Curriculum 32
3.3 Details of the MSc Program....................36
3.4 MSc Committees: Supervisory and Oral Examination....................36
3.5 Issues for the MSc Program....................37
3.6 Details of the PhD Program....................38
3.7 PhD Committees: Supervisory and Oral Examination 39
3.8 Issues for the PhD Program....................39
3.9 Recruitment, Admissions, Awards....................39
3.10 The Overall Process of Admission 41
3.11 Financial Support....................42
3.12 The Budget from Graduate Studies 45
3.13 The Selection of UVic Fellowships at Admission Time 46
3.14 Factsheet: Data on Graduate Students....................47
3.15 Graduate Co-op Program 49
3.16 Administrative Infrastructure....................50
3.17 Graduate Student Involvement 50
3.18 Social Activities and Collegiality: the Students' Views....................51

Chapter 4: Research and Scholarship 53
4.1 Faculty Research Interests 54
4.2 New Faculty Appointed in 2006....57
4.3 Adjunct Faculty, Cross-Listed Faculty and Professors Emeritus....57
4.4 Research Grant Funding 58
4.5 Research Collaborations 61
4.6 Undergraduate Involvement in Research 62
4.7 Publications and Other Measures of Research Impact....63

Chapter 5: Ancillary Activities 65
5.1 Equity 65
5.2 Outreach and Recruitment 66
5.3 Professional Activities....68

Chapter 6: Administrative and Teaching Support 75
6.1 Departmental Office Support Staff....75
6.2 Administrative Support Staff 76
6.3 Undergraduate Teaching Support Staff....76
6.4 Technical Support Staff 77

Chapter 7: Resources and Facilities 79
7.1 Personnel....79
7.2 Budget....81
7.3 Space 82

Appendix A: The North American Context A.1
A.1 Faculty A.1
A.2 Undergraduate Students.... A.1
A.3 Graduate Students A.3

Appendix B: Publications by Regular Faculty (2000–2007) B.1
B.1 Refereed Journal Publications 2000-2007 .... B.1
B.2 Refereed Conference Publications 2000-2007 .... B.9

Appendix C: Papers Published with Undergraduates C.1

Appendix D: Involvement of Adjunct Professors D.1
D.1 UVic Co-op Students and B. Sc. Graduates D.1
D.2 Masters Graduates.... D.2
D.3 Ph. D. Graduates.... D.2

Appendix E: Combined Programs E.1
E.1 History of Combined Programs E.1
E.2 Global Issues.... E.1
E.3 The Geomatics Program E.2
E.4 The Music and Computer Science Program E.3

Appendix F: Software Engineering F.1
F.1 The BSENG degree F.1
F.2 The Software Engineering Option in Computer Science F.4

Appendix G: Laboratories for CSC and SENG courses G.1

Appendix H: Two Proposals for New Options H.1

H.1 Networking Option H.1
H.2 Graphics Options H.2
H.3 Ensuring Success of Newly-Introduced Options H.5

Appendix I: Service Courses — Observations and Challenges I.1

I.1 History I.1
I.2 Observations I.1

Appendix J: Calendar Descriptions of Programs J.1

J.1 UVic/Malaspina University College Joint BSc; BSENG Program J.1
J.2 Computer Science Undergraduate Course Descriptions J.12
J.3 SENG Undergraduate Course Descriptions J.15
J.4 Graduate Programs J.17
J.5 Graduate Course Descriptions J.20

List of Tables

TABLE 2.1: Trends in enrollment for first year courses 18
TABLE 2.2: Course population grouped by course levels 19
TABLE 2.3: All degrees awarded with CSc designation from 1980 to 2006 21
TABLE 2.4: Headcount of BSc (declared) in the Faculty of Engineering 23
TABLE 2.5: Number of CSc degrees (Major and Honours only) 24
TABLE 2.6: Co-op Program: enrollment (winter term) and work term placements 26
TABLE 2.7: Computer Science/Math Co-op Program Enrollment 27
TABLE 2.8: Evaluation for "Overall Teaching Ability" 29
TABLE 3.1: Courses for Graduate Students Offered from 1999 to 2006 32
TABLE 3.2: Topics Courses Offered from 1999 to 2006 33
TABLE 3.3: Split numbering of 4XX/5XX level courses 35
TABLE 3.4: MSc Committees 36
TABLE 3.5: PhD Committees 39
TABLE 3.6: Admission to the MSc Degree 40
TABLE 3.7: Admission to the PhD Degree 40
TABLE 3.8: Standard Support (from September 2006) 43
TABLE 3.10: NSERC Support (from September 2006) 44
TABLE 3.11: Graduate financial support from 2001 to 2007 44
TABLE 3.12: RA and TA support as percentage of total support for MSc 44
TABLE 3.9: UVic Fellowship Support (from September 2006) 44
TABLE 3.13: RA and TA support as percentage of total support for PhD 45
TABLE 3.14: Contributions from Graduate Studies 46
TABLE 3.16: Graduate Degrees from 2000 to 2006 48
TABLE 3.15: A Snapshot in February 2007 48
TABLE 3.17: Graduate enrollment 49
TABLE 3.18: Graduate degrees with co-op and workterms 50
TABLE 4.1: Primary Research Areas of Research Faculty 54
TABLE 4.2: Adjunct Faculty for 2005/2006 58
TABLE 4.3: Cross-Listed Faculty for 2005/2006 58
TABLE 4.4: Professors Emeriti for 2005/2006 58
TABLE 4.6: Research Grants and Contracts, excluding Discovery Grants, 1999-2005 59
TABLE 4.7: Equipment Grants, 1999-2005 59
TABLE 4.5: NSERC Discovery Grants, 1999-2006 59
TABLE 4.8: Publication Counts for Faculty 63
TABLE 7.1: Course sections taught per year (May 1 – April 30) 80
TABLE 7.2: Operating funds and their categories 81
TABLE 7.3: Comparison of operating budgets with other UVic units 82
TABLE 7.4: Statistical Data on Space Designation 83
TABLE 7.6: Space Information (Between 2004 and 2006) 84
TABLE 7.7: Space Information (Before 2003) 84
TABLE 7.5: Space Information (Since May 2006) 84
TABLE A.1: Faculty members in CSc: UVic and Canada A.1
TABLE A.2: Faculty demographics in CSc: UVic and Canada A.1

TABLE A.3: CSc Degrees: UVic and North America A.2
TABLE A.4: Enrolment in Masters and Doctoral Programs A.3

List of Figures

FIGURE 2.1. 4th Year Prerequisite Tree. 11
FIGURE 2.2. The prerequisite tree for the 1st and 2nd Years 12
FIGURE 2.3. Graphs of course population by course levels 19
FIGURE 2.4. Graph of total course population 20
FIGURE 2.5. All Degrees Awarded with CSc Designation....22
FIGURE 2.6. BSc Major degrees from 1999 to 2006 (plotted from Table 2.5) 25
FIGURE 2.7. BSc Honours degrees from 1999 to 2006 (plotted from Table 2.5)....25
FIGURE 2.8. Computer Science/Math Co-op: enrollment and placements 26
FIGURE 2.9. Enrollment (for each term) by degree program 27
FIGURE 3.1. The requirements of the Graduate programs.....31
FIGURE 3.2. The admission process 41
FIGURE 3.3. The process of selection for UVic Fellowships 46
FIGURE 3.4. Graduate degrees awarded from 2000 to 2006 48
FIGURE 4.1. Major Research Areas in the Department (July 2006) 53
FIGURE 4.2. Average 2005/2006 Discovery Grant compared by University 60
FIGURE A.1. HERI/UCLA Study A.2

1: Overview

1.1 History of UVic and the Computer Science Department

The University of Victoria came into being on July 1, 1963, but it had enjoyed a prior tradition of sixty years teaching at the university level as Victoria College. This sixty years of history falls into three stages.

Between the years 1903 and 1915, Victoria College was affiliated with McGill University. The opening of the University of British Columbia in 1915 obliged Victoria College to suspend operations in higher education.

As a result of local demand, Victoria College began its second stage of development, reborn in affiliation with the University of British Columbia. In 1921, the College moved to the Dunsmuir mansion, Craigdarroch. Here, Victoria College built a reputation over the next two decades for thorough and scholarly instruction in first and second year Arts and Science.

The final stage, between the years 1945 and 1963, saw the transition from two year college to university. In 1946, post-war enrollment pressure resulted in the move of the College from Craigdarroch to the Lansdowne campus, now the site of Camosun College. The College, still in affiliation with UBC, awarded its first bachelor's degrees in 1961. Late in the transition period, the 284 (now 381) acre campus at Gordon Head was acquired. The University was born with the Universities Act of 1963.

Although the Department of Computer Science did not come into existence until 1980, the development of computer science courses at UVic began in 1963. In 1964, the University offered its first credit course in computer programming, and the next year added a numerical analysis course.

In 1969, the Computing Science Group was formed within the Department of Mathematics. The group consisted of four professors (I. Barrodale, S. Clarke, B. Ehle and F. Roberts). They organized the first comprehensive set of computer science courses at the University, and set up a BSc program in Mathematics with a computer science emphasis. The first students of this option graduated in 1972.

By the mid 1970's, the Universities Council of B.C. gave authorization for a BSc degree in computer science. Soon thereafter, the computer science group was sufficiently large that forming a separate department was beneficial. On July 1, 1980, the Department of Computer Science was formed with a complement of nine faculty, all with PhDs in computer science. The Faculty of Engineering came into existence in 1983, and in 1988, the Department of Computer Science was formally transferred from the Faculty of Arts and Science to the Faculty of Engineering.

The development of a graduate program in computer science at the University of Victoria paralleled that of the undergraduate program. In 1970, an MSc degree program in computer science was established within the Department of Mathematics, and between 1971 and 1976 approximately 25 MSc degrees were awarded. The late seventies saw an emphasis on a broader Applicable Mathematics degree. Beginning in 1981, a graduate program in computer science was reactivated and a formal PhD program was approved in 1988. Since the department was formed, approximately 1 M.A., 280 MSc and 58 PhD degrees have been conferred.

Following a sharp fall in enrolment in the early nineties, numbers picked up again so that by 1998, numbers were again at record highs, and began the very rapid increase that was experienced by all departments in North America, leading to a doubling of the number of degrees granted in 2004 as compared with 1998 and four times the number compared to 1990.

In 2001, the Provincial Government introduced the *Doubling the Opportunity Program (DTO)* designed to sharply increase the number of graduates in computer science and electrical and computer engineering. The introduction of this program coincided with the beginning of a fall in enrolment in computer science across North America. Eventually the overall numbers decreased to about half the level of the previous peak on average. While UVic has not had reductions as steep as the average level, it has meant that the increases envisaged under DTO have not yet been achieved, and it will likely be another five years before we achieve these levels. The graduate program has increased in size considerably in the last few years, so that it is now the largest thesis-based program at UVic.

The undergraduate program offerings of the department have expanded considerably in the last five years with the introduction of a number of innovative combined degrees, with the first graduates from the combined program with Visual Arts convocating in 2006. In addition, the Bachelor of Software Engineering program, run jointly by Computer Science and Electrical and Computer Engineering was introduced in 2003. The department now offers 19 distinct undergraduate programs.

1.2 Purpose of the Department

It is the mandate of each academic unit within the University to develop and implement educational programs and to promote scholarship and research within its discipline. The following section describes, in broad terms, how this mandate is being implemented by the Computer Science Department.

The department continues to assess and evolve its undergraduate and graduate programs to ensure the excellence of its graduates. It is important to achieve a balance between graduates prepared to make an immediate contribution to the needs of employers in industry, government, and academia. A computer science undergraduate program should take into account current practice. It is also important that the curriculum be challenging. Our co-operative education programs constitute a major facet of our undergraduate program.

The purpose of a graduate and research program is to advance the state of knowledge in selected areas of computer science, to generate both theoretical and practical research results, to contribute significantly to the training of highly qualified personnel, to collaborate with other universities and industrial partners, to work actively on technology exchange with government and industry, and to make other contributions to the research community at large such as organizing conferences.

Another important role of the Computer Science Department is to promote computer literacy and computer competence both within and outside the University by providing a variety of educational programs and courses.

1.3 Department Assessment

The Department of Computer Science at the University of Victoria has evolved dramatically over the past ten years. In this period, our department has grown from 22 to 34 faculty members. In the same time frame the number of undergraduate degrees has increased from 65 to 126 last year, while our graduate student numbers have increased from 65 to 120. Over the past decade, very strong and innovative research programs in diverse areas have been established producing excellent research results as well as outstanding graduate students. The research funding level of the department has increased dramatically in recent years and is now at a reasonable level. However, considering the current federal and provincial funding climate, it is anticipated that there will be ongoing challenges for adequate funding for both teaching and research.

The last external review was undertaken in 1995 at a time when the department had 21 faculty members, and awarded 66 bachelors degrees and 15 postgraduate degrees. This compares to 31 faculty members, 114 bachelors degrees and 33 postgraduate degrees last year.

The department is committed to excellent teaching, continual updating of our curriculum, and regular teaching evaluation. We recently (May, 2006) moved into a new building (ECS) which includes all of our teaching facilities, our research laboratories and all of our graduate students. Thus the space separation problems that existed for a number of years have been solved.

The administration of the department, including graduate student recruitment and co-op programs, has been running smoothly and effectively for some time. There are many people who are working extremely hard behind the scenes to make this a reality.

A few years ago, the University embarked on *Project Nova*, which is an ongoing initiative to replace the University's technologically-dated and stand-alone systems with a state-of-the-art information technology infrastructure. The project is part of the University's Strategic Plan and makes UVic one of the first universities in North America to move to a "unified digital campus." Project Nova encompasses all aspects of the operation of the university, including new systems for finance, facilities, human resources, student records and scheduling. It is going to have a major impact on the way in which units carry out their mandate, but as yet, we are unsure as to the exact implications.

We all believe that the Department of Computer Science at the University of Victoria constitutes an excellent and exciting working environment.

2: Undergraduate Education

2.1 Overview

Computer Science courses have been delivered at the University of Victoria since 1964. As our discipline has grown and evolved, so has the range of our programs offered to the undergraduate students. The first CSc program was a BSc in Mathematics with a Computing Science emphasis, offered in the Department of Mathematics in 1969. The BSc Major in Computer Science was introduced in 1979. We now offer a wide range of Combined Programs and we offer a Bachelor of Software Engineering degree (BSENG) in partnership with the Department of Electrical and Computer Engineering. The changes in our programs reflect the changing role of IT in society.

The curriculum changes have taken place within an environment that supports teaching of high quality. CSc students have had great success in obtaining jobs both within our cooperative education program and upon graduation. Many have chosen to continue their studies at the graduate level, and many of these have obtained NSERC postgraduate awards. Recent graduates have enrolled in graduate studies at prestigious universities, including Carnegie Mellon, Cornell, Princeton, Stanford, UBC, Toronto and Waterloo.

In this section, we focus on the three main groups of CSc programs offered:

- **"Traditional" CSc programs**: These are non-interdisciplinary programs comprising the BSc Major in Computer Science, BSc Honours in Computer Science, BSc General in Computer Science, and BA General in Computer Science. Also included here is a Minor in Computer Science and the Co-operative Education Program.
- **Interdisciplinary Programs**: Throughout our department's history we have acknowledged the role that computer science plays in other disciplines. Our interdisciplinary programs are, by design, different from double-major degrees. We offer two types: *Options* and *Combined Degrees*.
  The Options programs are in the areas:
  - *Bioinformatics* (with the Department of Biology),
  - *Business Administration* (with the Department of Business Administration),
  - *Software Engineering*,
  - *Mechatronics & Embedded Systems*.

  The Combined Degree programs combine CSc with (in alphabetical order):
  - *Geography* (with the Department of Geography); this combination is also known as "*Geomatics*",
  - *Health Informatics* (with the Department of Health and Information Science),
  - *Mathematics* (with the Department of Mathematics and Statistics),
  - *Music* (with the School of Music),
  - *Physics* (with the Department of Physics),
  - *Psychology* (with the Department of Psychology),
  - *Statistics* (with the Department of Mathematics and Statistics),
  - *Visual Arts* (with the Department of Visual Arts).

- **Software Engineering and BSENG:** The software engineering option in Computer Science is substantially different from the professional engineering BSENG program. Computer Science is responsible for delivering about 70% of the BSENG program.

Our department therefore offers 18 different CSc programs involving (and requiring coordination with) 11 other academic units at the University of Victoria. We believe this complexity to be unique to our department. However, our choice to provide so many choices to undergraduate students is itself a strategic response to the evolution of our own discipline.

2.2 "Traditional" CSc Programs

BSC MAJOR IN COMPUTER SCIENCE

The *Major* program is the central degree program for students who plan a career in CSc. The program is composed of:

- two general first-year CSc courses,
- three second-year CSc courses and one SENG course,
- seven third-year CSc courses,
- a choice of three fourth-year CSc courses,
- math courses at the first- and second-year levels, a second-year statistics course, two writing courses, and open electives (i.e., no restrictions on choice of electives beyond year level).

To graduate "With Distinction", a graduating GPA of 6.5 (i.e., B+/A-) in the program and in all courses in the department numbered 300 or higher must be obtained.

Within the BSc major degree, students can specialize in a particular area in the upper years (Areas of Emphasis). Current areas of emphasis are Algorithms, Programming Methodology, Scientific Computing, Systems, and Software Engineering.

BSC HONOURS IN COMPUTER SCIENCE

The *Honours* program is the premium CSc degree. It is well suited for students who plan to continue into graduate studies. An Honours student must achieve a minimum GPA of 5.0 (i.e., a B) to graduate with an Honours degree. To graduate "With Distinction", a graduating GPA of 6.5 (i.e., B+/A-) in the program and in all courses in the department numbered 300 or higher must be obtained.

BSC AND BA GENERAL IN COMPUTER SCIENCE

The General program provides the opportunity for students to include CSc as part of a more general BSc or B.A. program. Students in this program take the same CSc, Math, Statistics and SENG courses in their first and second years as students in the Major program. Afterward, they take at least six third- and fourth-year CSc or SENG courses within their degree program.

MINOR IN COMPUTER SCIENCE

Students from other departments can obtain a Minor in CSc. The requirements are similar to the requirements for the BSc and BA General in Computer Science, but with a slight reduction in the math requirements.

2.3 Interdisciplinary Programs

The two categories of interdisciplinary programs are *Option Degrees* and *Combined Degrees*. Their philosophy can be summarized as follows.

- The main objective of an *Option Degree* is to provide the student with an overview of the complementary field of study, while giving the most emphasis to CSc.
- *Combined Degree* programs consider both CSc and the complementary field to be equals; the student is expected to gain a similar level of expertise in both fields.

Another very important difference between the two types of programs is that Option Degrees are granted by the Faculty of Engineering, while the Combined Programs Degrees are granted by the Faculty of the collaborating academic unit. For example, graduates from the *Geomatics* combined program obtain a Bachelor of Science degree granted by the Faculty of Science. Furthermore some programs, such as *Visual Arts* and *Music*, give the student the option to receive a Bachelor of Arts or a Bachelor of Science. Appendix E provides a chronology of program introductions.

Here are some benefits we believe these programs bring *to students*:

- The CSc graduates have a wide range of job opportunities and in a variety of fields.
- Students have more educational options when pursuing a CSc degree at UVic. For example, they can choose between a broad CSc degree (such as a Major in CSc), or one with a more formal focus on some other domain as well (such as one of the combined programs), without the rigorous expectations of a Double Major degree.

Some benefits of these programs *to the department* are:

- The combined programs attract students who would not have considered doing a traditional major in CSc, enhancing our diversity.
- The new programs have enhanced the relationship between CSc and other departments at the university. These relationships are both academic (via cross-listed courses in both departments) and research-oriented (through collaborative research projects).
- Our department has gained a reputation as an innovator in interdisciplinary education in Canada.

We believe that these programs bring benefit *to the University of Victoria* because:

- The uniqueness of these programs attracts students who may not have otherwise considered attending this university.
- The programs enhance relationships between different academic units.

In the rest of this section we list the *Options* and *Combined Degree* programs which are offered. Additional notes for all combined programs can be found in Appendix E.

BIOINFORMATICS OPTION

This program covers techniques in mathematics, information science, statistics and CSc used to solve problems in biology. It is targeted to those interested in research and technologies in bioinformatics such as modeling evolution, dating evolutionary events, understanding biological sequences and assisting medical research in the understanding and elimination of certain diseases, and the detection or correction of genetic defects.

BUSINESS ADMINISTRATION OPTION

Students in this option have the opportunity to enroll in upper-level business classes that are usually reserved for Faculty of Business students. Graduates from this program therefore have a strong background in both CSc and business and are perfectly positioned to exploit today's IT/business relationships. The option includes a mandatory Co-op (as expected by the Faculty of Business).

MECHATRONICS & EMBEDDED SYSTEMS (MES) OPTION

This newer option integrates mechanical, electrical/electronic control and computer systems into products and processes, the students can use real-time computation techniques while building intelligent electro-mechanical systems. The merging of multiple disciplines plus teamwork skills needed to work effectively while building such systems helps produce the kind of engineer or computer scientist who is currently in high demand. The option includes a mandatory Co-op.

COMBINED PROGRAM IN GEOGRAPHY AND COMPUTER SCIENCE (GEOMATICS)

Geomatics combines cartography, CSc, Geographic Information Systems, remote sensing, spatial analysis and surveying. Students with these skills work in government agencies, forestry, agriculture, mining, surveying, resource management, oceanography, hydrography, environmental management, urban planning, public utilities, coastal zone management, transport, navigation, defense, medical epidemiology, and pollution management (to name several areas).

COMBINED PROGRAM IN HEALTH INFORMATION SCIENCE AND COMPUTER SCIENCE

Over the next decade, Canada will need thousands of highly qualified personnel trained on the intersection of computer science and health sciences to respond successfully to the challenges of an aging population and increasing health care costs. This program addresses these needs. It is the combined program with the highest enrollment. It attracts students originally interested in the Health and Information Sciences degree, yet wanting a stronger and more in-depth technical background. The program includes a mandatory co-op.

COMBINED PROGRAM IN COMPUTER SCIENCE AND MATHEMATICS

This is one of the oldest programs in the department. The program usually attracts the academically strongest students, many of whom have gone on to graduate school and careers in research and academia.

COMBINED PROGRAM IN MUSIC AND COMPUTER SCIENCE

This program enables students to join their interest and knowledge in Music with Computer Science and technology. It is a good choice for students who have an interest in music, yet cannot or do not wish to pursue a Music Performance (BMus) degree. Graduates will be able to work in areas involving music recording and production, creating music for films and video games, and other parts of the entertainment industry. The program includes an optional co-op.

COMBINED PROGRAM IN PHYSICS AND COMPUTER SCIENCE

The program started in 1998. Similarly to the combined program with mathematics, it usually attracts the academically strongest students.

COMBINED PROGRAM IN PSYCHOLOGY AND COMPUTER SCIENCE

Psychological concepts are important in designing effective user interfaces for systems, for understanding and working with artificial intelligence, in methods of biological perception, in determining neural modes of computation, and in modeling and understanding of cognitive functions. Careers for students with the skills obtained in such a combined degree can be found in software and hardware design, HCI, robotics, medical technology, biometrics and psychometrics, evaluation and rehabilitation, artificial intelligence systems, and research in human cognition. The program has an optional co-op.

COMBINED PROGRAM IN COMPUTER SCIENCE AND STATISTICS

This program is one of the oldest in our department. It graduates a small number of students every year. In a similar manner to other combined programs such as Math+CSc and Physics+CSc, students need not indicate interest in the program until they have completed several terms at UVic, since many courses are in common between the degrees.

COMBINED PROGRAM IN VISUAL ARTS AND COMPUTER SCIENCE

The computer has become a major tool and medium for the creation and display of visual arts. The objective, therefore, is to prepare students to become the new mediators between the digital art and computer science domains. Students with these skills may work in the entertainment industry, in art galleries and museums, in cultural heritage, in marketing and promotion, in multimedia design and production, or as designers and project managers. Several students have graduated from the program and have been successful finding jobs in their field. The program has an optional co-op.

2.4 Software Engineering

Software Engineering is the most popular degree option of all those available in the department. A new alternative is provided by the formal Bachelor of Software Engineering (BSENG) degree, which is offered in cooperation with the department of Electrical and Computer Engineering. Both types are further detailed in Appendix F.

SOFTWARE ENGINEERING OPTION

Software engineering combines CSc with engineering principles and best practices to design, implement, test, maintain, and develop software. The Option also gives students knowledge into the management components of software engineering. An Option is more concentrated than an Area of Emphasis.

BACHELOR OF SOFTWARE ENGINEERING DEGREE (BSENG)

The Bachelor of Software Engineering (BSENG) program has been designed to provide a solid and industrially relevant engineering education with focus on software engineering. BSENG students also acquire communication and business skills and become proficient in engineering design, particularly as it applies to software development and software systems. Through mandatory co-op, students have the opportunity to participate in one of the largest experiential learning programs in the country. The program is designed to be accredited by both the Canadian Engineering Accreditation Board (CEAB) and the Computer Science Accreditation Council (CSAC). The CEAB accreditation is currently ongoing. The Faculty went through the CEAB site visit in January 2007. A complete description of the program, including explanations of its curriculum and planning, may be found in Appendix F.

2.5 Service Courses

We have five main service courses in CSc:

1. **CSC 100** (“Elementary Computing”) - It is used in several departments as either an elective or as a course that is strongly suggested.
2. **CSC 105** (“Computers and Information Processing”) -This course has the additional restriction that, if taken by someone who later decides to enroll into a Computer Science Major degree program, it cannot be counted towards the degree, not even as an elective. This course is required by Economics and priority for seats is given to these students. It is also used as an elective in several departments and is often taken by students before they enter a program in Business. The curriculum is developed in consultation with the various user departments.
3. **CSC 110** (“Fundamentals of Programming I”) - This is a required course for all Computer Science degrees, but it is also a service one since it is required for Engineering, Math, Statistics, Physics, Astronomy, and Health Information Science.
4. **CSC 115** (“Fundamentals of Programming II”) - This is a required course for all Computer Science degrees, but it is also a service one since it is required for Engineering, Math, Statistics, Physics, Astronomy, and Health Information Science.
5. **CSC 160** (“Fundamentals of Programming II for Engineers”) - This is often similar to CSC 115. It is a required course for students in a B.Eng. program. The CSC 160 course material is developed in consultation with the other two departments in Engineering. This course is now being phased out and replaced by the new CSC 111 for the Engineering programs, to be offered for the first time in Fall 2007.

Service courses are normally designed through consultation with other departments — typically the ones for whom we are providing the course as a service. CSC 100 is slightly different, as other departments do not work directly with us to set the curriculum. However, communication and consultation between the CSC 100 “course champion” and members of other departments takes place on a regular basis.

For the most part, communication between different departments and Computer Science on the topic of curriculum for service courses is very collegial. Problems can occur when the service course at some point does not meet, or is perceived not to meet, the needs of the outside department, and curriculum renewal needs to take place through consultation. Further history, observations and challenges raised by our service courses can be found in Appendix I.

2.6 Types of Courses and Prerequisites

The diagrams in Figures 2.1 and 2.2 show the diagrams of the prerequisite trees.

FIGURE 2.1. 4th Year Prerequisite Tree.

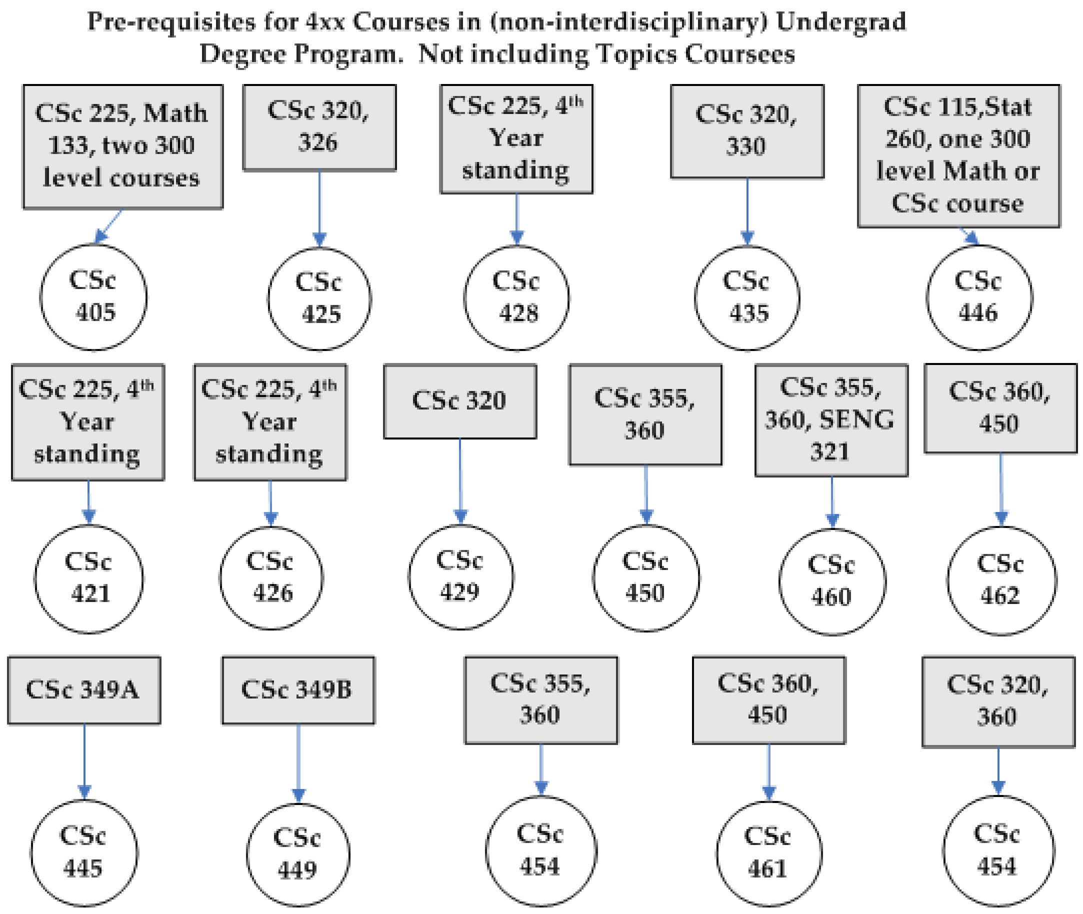
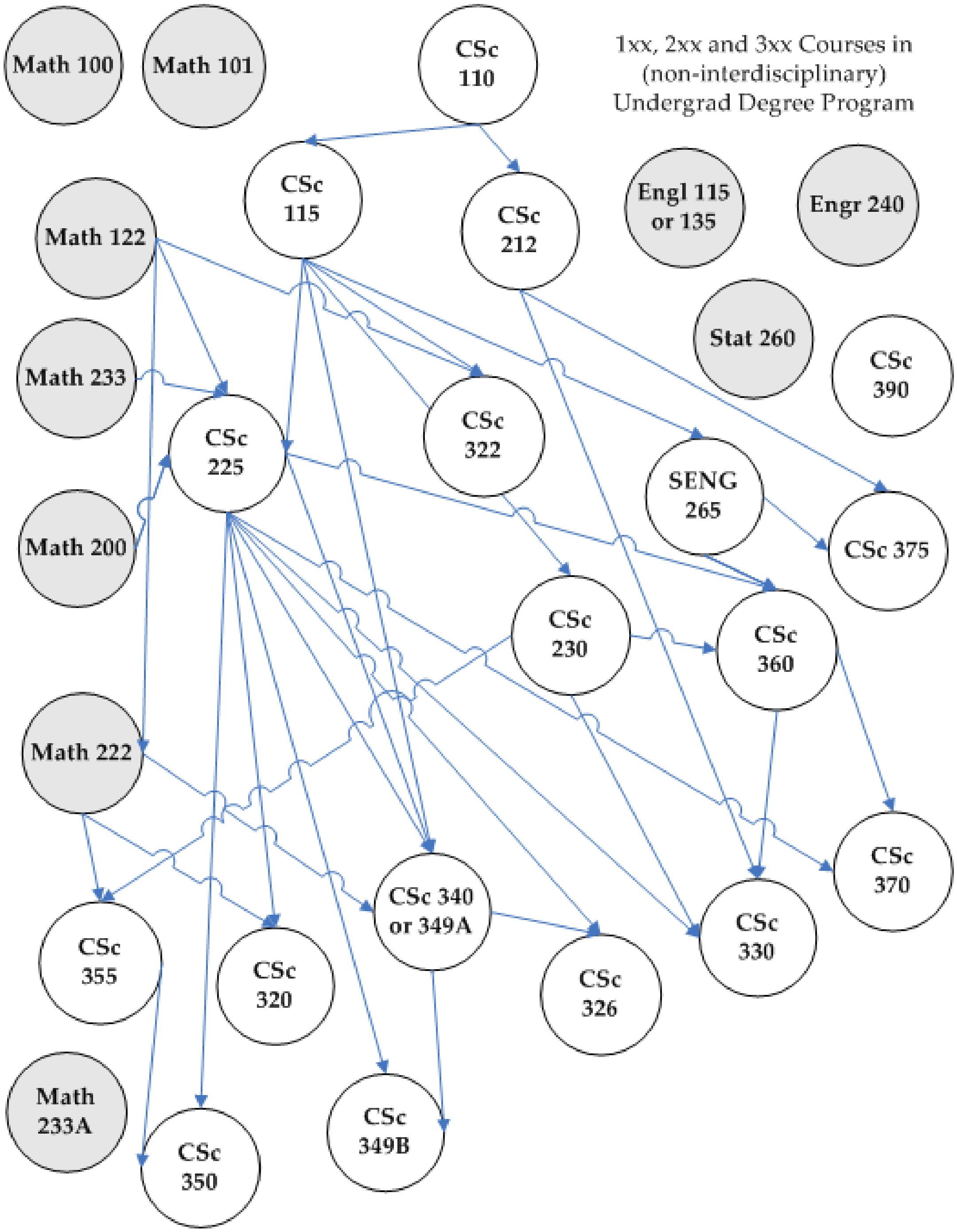

Pre-requisites for 4xx Courses in (non-interdisciplinary) Undergrad Degree Program. Not including Topics Coursees
CSc 225, Math 133, two 300 level courses
CSc 405
CSc 320, 326
CSc 425
CSc 225, 4th Year standing
CSc 428
CSc 320, 330
CSc 435
CSc 115,Stat 260, one 300 level Math or CSc course
CSc 446
CSc 225, 4th Year standing
CSc 421
CSc 225, 4th Year standing
CSc 426
CSc 320
CSc 429
CSc 355, 360
CSc 450
CSc 355, 360, SENG 321
CSc 460
CSc 360, 450
CSc 462
CSc 349A
CSc 445
CSc 349B
CSc 449
CSc 355, 360
CSc 454
CSc 360, 450
CSc 461
CSc 320, 360
CSc 454

FIGURE 2.2. The prerequisite tree for the 1st and 2nd Years



Beyond the regular courses at all levels as listed in the calendar, together with their accompanying labs or tutorial sections, there are two other types of courses which can be offered to students, normally at the 4th year level.

2.6.1 TOPICS COURSES

The department offers a variety of advanced subjects as “topics courses”. These courses tend to cover ‘hot’ topics in a particular field of study and typically relate to professors’ areas of research. If a topics course shows sustained demand, it can become a regular course offering, after the approval of the department through the normal curriculum revision route. Some areas for topics courses in recent years have been:

- Advanced Networking
- Firewall and Security
- Music Information Retrieval
- Optimization
- Wireless Networks
- Computational Biology
- Graph Algorithms
- Pervasive Computing
- Randomization
- Hypermedia
- Component Software
- Data Compression
- Virtual Machines
- Databases
- Embedded Systems
- e-Commerce
- Combinatorial Algorithms.
- Visualization
- Artificial Intelligence
- Network Reliability
- System Infrastructure Design
- Computational Complexity

2.6.2 DIRECTED STUDIES COURSES

Sometimes, courses are offered by faculty in the form of *directed studies*, where only one or a very small number of students are given instruction in a specific field. This usually takes place when enrollment may not warrant scheduling a topics course. Each offering must be approved by the Chair or the Director of Undergraduate Studies. Directed studies are normally related to a specific research area or simply as a tool for advanced learning. Some areas for directed studies courses in recent years have been:

- Advanced Networking
- Software Development
- Computer Music and related software projects
- System analysis
- Advanced Computer Architectures
- Performance Evaluation.

2.6.3 HONOURS PROJECT (CSC 499)

A special type of Directed Studies course is restricted to students in the Honours program. The course is supervised overall by the instructor who is also in charge of the Graduate Seminar. However each student is paired with a faculty member for a challenging capstone project which is intended to give a preview of research activities.

2.6.4 LABORATORY WORK

As part of the practical training component of courses, the Computer Science department offers regularly scheduled labs in courses at all years. Appendix G provides a list of the courses which have regularly scheduled labs each week.

2.7 Curriculum Renewal

Recognizing the trends and developments in the field of computer science, the department reviews the course offerings frequently and updates the curriculum.

2.7.1 NEW COURSES (AND OPTIONS) IN GRAPHICS AND NETWORKS

Major curriculum changes are currently proposed in the areas of Graphics and of Networks. In both areas, some of the 4th year courses are to be moved to 3rd year, and new 400 level courses introduced. A new Graphics Option program and a new Networks Option program will be available and details can be found in Appendix H.

2.7.2 COURSE-LEVEL REDEVELOPMENT

In the past seven years the following courses have received major review:

- CSC 110: Yvonne Coady and LillAnne Jackson headed an instructional team to redevelop the course, focusing on improving the first year students' abilities to solve problems.
- CSC 115: Piotr Kaminski, Hausi Müller and Peggy Storey redesigned the course in 2000-2003.
- CSC 115: The team looking at CSC 110 also developed new learning objectives for this course, resulting in all new assignments and labs, developed by Jason Corless, in conjunction with the lab leader.
- CSC 160: Jason Corless and Dan Hoffmann completely redeveloped the course, to focus on providing a more complete understanding of the procedural programming for engineers.
- CSC 230: Nigel Horspool and Micaela Serra headed an instructional team to redesign the course, with a change of emphasis towards more architectural issues and a new practical platform (the ARM). This entailed the development of an ARM simulator and the introduction of new connected lab boards.
- CSC 225: Ulrike Stege received a Learning and Teaching Centre (LTC) grant for the project "Teaching Algorithms and Data Structures with Web Animations: Application to CSC 225 and Other Interdisciplinary Applications". This was the basis for redevelopment for CSC 225 to include more applications.
- CSC 355: LillAnne Jackson made significant changes to the hardware labs in the course, introducing a much higher proportion of programming in a hardware language (namely VHDL), using state-of-the-art commercial development tools for circuit design (the Xilinx suite).
- CSC 370: Daniel German received an LTC grant for "Evaluating the use of c-spatia and Scale-up in CSC 370 Database Systems".
- CSC 405: Brain Wyvill undertook a major revision of the course for the January 2007 offering.
- CSC 428: Ulrike Stege received an LTC grant for "A template for interdisciplinary, modeling-based computer science courses".
- CSC 450: Dan Hoffman received an LTC grant for "RouterLab: Hands-on study of the internet infrastructure", where he developed a full software platform to be used in CSC 450, together with a full sets of labs.

- Jens Weber-Jahnke has just received an LTC grant for "Problem-based Learning Patterns for Practical Computer Science and Software Engineering Courses": *to develop, introduce and evaluate patterns of problem-based learning into practical courses. Problem-based learning is a learning approach that is an alternative to the traditional style of introducing learning material first and then exercise them with assignments. In PBL, problems come first and drive the introduction of learning material.*

2.7.3 PROGRAM LEVEL REDEVELOPMENT

A mechanism chosen in 1995 by the department to support both curricular renewal and monitoring was the concept of "Course Champions". Each of our undergraduate core classes (CSC 110, CSC 115, CSC 212, CSC 230, CSC 265 and CSC 225) had assigned to it a set of champions, with one person heading such a group of faculty and teaching staff. The groups were tasked with preparing detailed course outlines for each course. One of the important goals of this mechanism was to ensure as much consistency between course offerings as possible, while at the same time still providing for some flexibility in topics to those assigned to teach these courses. Normally 80% of the material is prescribed by course champion groups, while 20% is left to the discretion of each instructor.

A combination of events starting in 1998 caused the mechanism to strain. In January 1998 the language of instruction of our CS1 course (CSC 110) moved from C to Java. A similar change was made to our CS2 course (CSC 115). Also at that time, and for a few years following, the demand for our courses increased dramatically as a result of the dot-com boom. This increase in demand did not immediately translate into more resources, as these came later as a result of the efforts of Dean Miller and the DTO. What we now know is that the switch to Java, that is, teaching object-oriented programming concepts, resulted in a significant change in the content of CSC 110. The need to meet student demand for our courses did not immediately expose the mismatch between what was covered in the first year courses and what instructors of second-year courses expected them to cover. Said simply, the outlines which pre-date 1998 were never appropriately updated to adjust to the CSC 110 changes. In this experience, our department is not alone; many other CSc schools across North America have reported similarly mismatched experiences.

Computer Science used to have a prerequisite grade requirement for the core first and second year courses, in that students required a minimum grade of B- in CSC 115, CSC 225, CSC 230 and SENG 265 to continue in courses that had these as prerequisites. This had been introduced partly to enhance quality and partly because of the pressure of student numbers. This requirement was abolished in 2004-05 when the enrolment pressures abated.

Since 1998 the taught syllabi of some of these core courses has fluctuated (i.e., has not necessarily been consistent from offering to offering). The beginnings of a solution to this has been a task force to reconsider the curriculum of the core first year courses. A small group has redeveloped CSC 110 and 115. Briefly, a major goal of the CSC 110 redevelopment is to reintroduce a problem-solving emphasis. For CSC 115, the efforts have been to return carefully to what are considered "the basics" (linked lists, stacks, queues, and recursion), while including a small, albeit appropriate, number of object-oriented concepts. The initial offerings of the redeveloped first-year curriculum courses will soon finish. In the following term, we will obtain measurements designed to indicate whether or not our efforts have succeeded in their goals.

We are now planning to follow a similar process in updating the outlines and contents for the other core classes, specifically in second year, in order to ensure the appropriate level of consistency and expectations amongst offerings of the same course. Such an effort will also help instructors who teach our lower-level courses, as it would give concrete guidelines on expected outcomes. As the second-year courses have not had a similar examination since the Course Champion exercise of 1998, it would appear that a such an examination is of the highest priority.

2.8 Co-operative Education

The Co-operative Education program in the Department of Computer Science began in 1977. It was one of the first in Canada. The program integrates classroom with workplace learning. Students are required to complete four work terms, which are four-month periods of paid discipline-related employment. The program is very flexible and thus it is possible for a student to take two or even three work terms in succession if that is required by the employer. The department supports this flexibility by offering key courses at the 200, 300 and 400 levels during *all* three terms of the year. Thus it is possible for students to take campus terms and work terms in various sequences.

The Computer Science/Math Co-op program is an optional program leading to a "Co-op" designation on each type of degree chosen. Students can apply to enter the program at various times during their first two years, and admission is based on grades in certain required computer science and mathematics courses in the degree programs. All co-op students receive instruction on communication skills, resumé writing, job search and interview skills prior to their first work term.

2.8.1 CO-OP FEATURES AND ORGANIZATION

Some of the best distinguishing features of our co-op compared to others in the country are the personal attention and advising for each student, the geographical span of employers and jobs, the interconnections and interdisciplinarity of the overall co-op at UVic, and its integration in the university fabric. The University of Victoria possibly has the most diverse set of co-op programs in Canada.

Each student is assigned to a co-op coordinator who functions as the primary adviser and meets often to help in academic and job-related decisions. Close contacts are maintained. For every job placement and for every student, a co-op coordinator does a *site visit*, no matter where the position is geographically located. During the visit, the coordinator meets with both the student and the direct supervisor, for a mid-term evaluation from all perspectives. This approach has maintained a high level of success and satisfaction for students, employers and the university, and we believe this is the best way to prevent problems. Remarkably few unusual situations have to be dealt with given the scope and the size of the program.

The site visits also ensure that the job description is followed and is appropriate for the goals of the academic program. At the end of each work term, a student is required to write a "Workterm Report" in the style of a regular academic or industrial technical report, showing the required level of communication skills together with the acquisition of knowledge and experience. Each report is read and evaluated by a faculty member or a co-op coordinator after being approved by the work manager. Much editing may be required over the span of a few weeks to make it acceptable and to obtain a passing grade for the co-op term. A student can fail a term of co-op work either by having to leave a job or by not being able to submit an appropriate technical report, and would not be able to continue in the co-op program. This level of attention is something which indeed requires some intensive commitment on everybody's part.

Since its inception, and partly for historical reasons, the Departments of Computer Science and Mathematics and Statistics have operated a joint co-op program. However, the staff in this program has recently been integrated with that of the Engineering Co-op program, to form the Engineering and Computer Science/Math Co-op Program. It has one program manager and is housed in the new Engineering Computer Science Building. The department maintains an active involvement in academic matters within the co-op program, with faculty members serving on the Admissions and Retention Committees and all faculty members assisting with the evaluation of the mandatory workterm reports of co-op students.

2.8.2 INNOVATIONS AND CHANGES

One major innovation is that, since 2005, the Computer Science/Math Co-op program has offered what is believed to be the first *Work Experience Program* at a Canadian university. Students completing two work terms and the requirements of any major or honours degree program in either department obtain the *Work Experience* designation on their degree. This program enables, for example, advanced students, who had not previously been formally in the co-op program, to obtain 8 months of work term experience just prior to graduation, thus receiving benefit from the co-op preparation program. Furthermore, the *Work Experience* program provides a larger pool of senior students available for job placements.

The period from 1998/1999 to 2006/2007 has been a period of significant change in the cooperative education program. In 1998 and the following few years, the Information Technology industry was thriving, resulting in the co-op program placing a record number of students in 2000/2001 (with increased pressure on all aspects). However, that year was also the start of the turnaround in the IT industry. Since then, the number of co-op placements has decreased. Significant factors in this regard have certainly been the decline in the job market and increased tuition fees for the co-op program. However, increased competition from other new co-op programs (especially in British Columbia), and, of course, the decrease in the number of students entering Computer Science in general has had an impact.

Data relevant to Co-op can be found in Sections 2.9 and 2.10 with the overall CSC numbers. The Computer Science/Math co-op program has identified declining enrolment as a significant problem. The introduction of the Work Experience Program has helped to increase student numbers, but one of our current goals is to attract an even larger percentage of the students in our degree programs into the co-op and work experience programs.

The Computer Science/Math co-op program has a high percentage of international and ESL students, namely 28%. The co-op program recently decided to provide extra support for these students, and a coordinator was provided with extra training in this field. Other professional development was also provided to other staff members, and the international/ESL student coordinator has been asked to organize workshops on interculturalism to the co-op staff as well. The first goal of the student support is to reduce the number of poorly written and badly organized resumés and cover letters, as well as to provide increased training for interviews. Students are also given assistance with navigating issues concerning work permits. This extra attention will, we hope, increase the placement rate in this group, as these are the students who are most likely to become frustrated and give up finding a job or decide not to continue in the program.

2.9 Enrolment Trends in Undergraduate CSc at UVic

2.9.1 STATISTICS

Course population is one of the most important pieces of data as it affects directly the learning and teaching environment.

The first trend in Table 2.1 shows the enrollment in first year courses from 2001/02 to the present, including the service courses not necessarily for CSc students (namely 100/105).

Note that CSC 110/115 are the required courses for all CSc degrees. Note also that the entries start exactly at the time of the major “bubble” encountered all over North America and thus the decrease may be perceived too strongly.

TABLE 2.1: Trends in enrollment for first year courses

	2001/02	2002/03	2003/04	2004/05	2005/06	2006/07
CSC100/105	1393	1142	1090	1018	1169	1293
CSC 110/115	1102	931	872	764	741	700

The overall course population is shown in Table 2.2, divided by academic years and by levels of courses. It is interesting to note the numbers as reported in the last program review, 12 years ago, shown in the last row of Table 2.2 for 1994/95.

TABLE 2.2: Course population grouped by course levels

	1st year	2nd year	3rd year	4th year	Total
2001/02	2804	1201	1993	547	6545
2002/03	2381	1133	2176	710	6400
2003/04	2298	1020	2057	943	6318
2004/05	2075	712	1547	700	5034
2005/06	2250	664	1262	569	4745
2006/07	2294	607	1203	494	4598
1994/95	*1515*	*632*	*791*	*290*	*3228*

The same data from Table 2.2 is shown also in the charts of Figures 2.3 and 2.4, so that the trends over the years are more readily apparent. I.

FIGURE 2.3. Graphs of course population by course levels

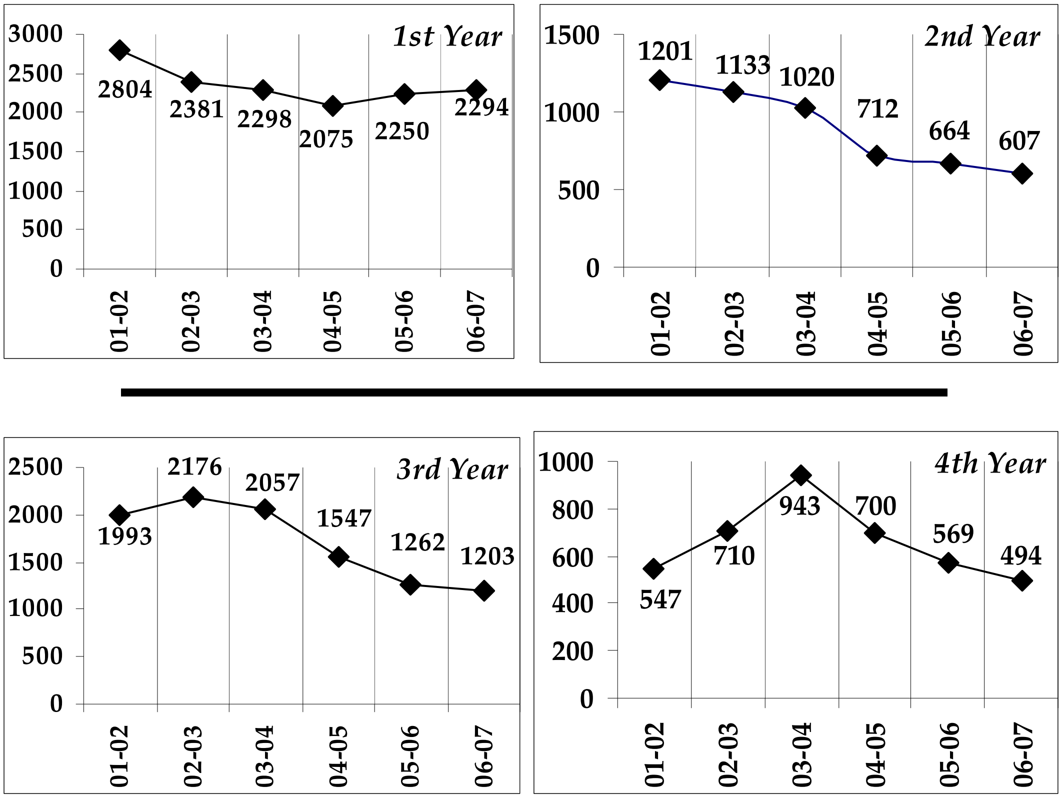
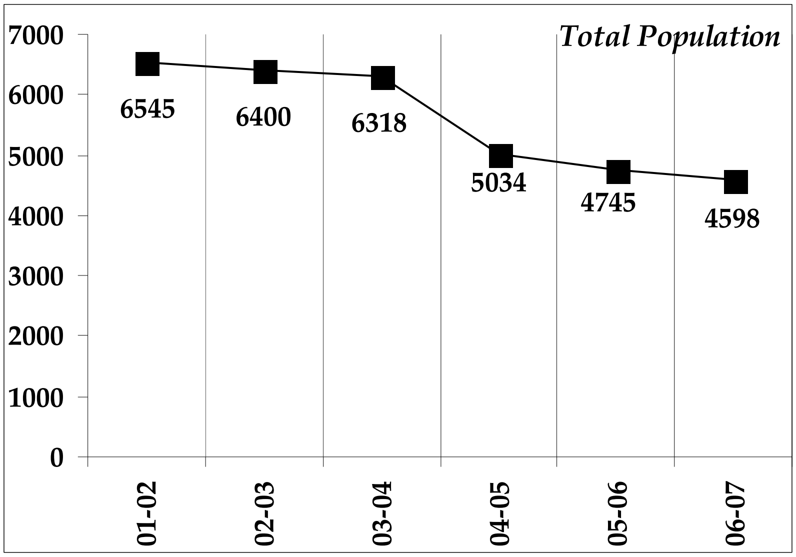


FIGURE 2.4. Graph of total course population



2.10 Student Demand and Performance

The data in the following tables are relevant to the topics included in this document. In some cases, charts are also given, as they show more clearly the trends.

First of all it is important to look at the total number of CSc degrees awarded over the span of many years. In the "CSc Degrees" designation here *all* degrees are included, that is, Major, Honours, Combined, with or without Co-op or Work Experience. Table 2.3 and the chart in Figure 2.5 give this data.

The cyclical nature of the demand for CSc programs is very clear from Figure 2.5, with the sharp rise in the early 80's. After 1984, it was another 14 years before the number of degrees returned to that level. Some of these pivotal years are highlighted in the graph.

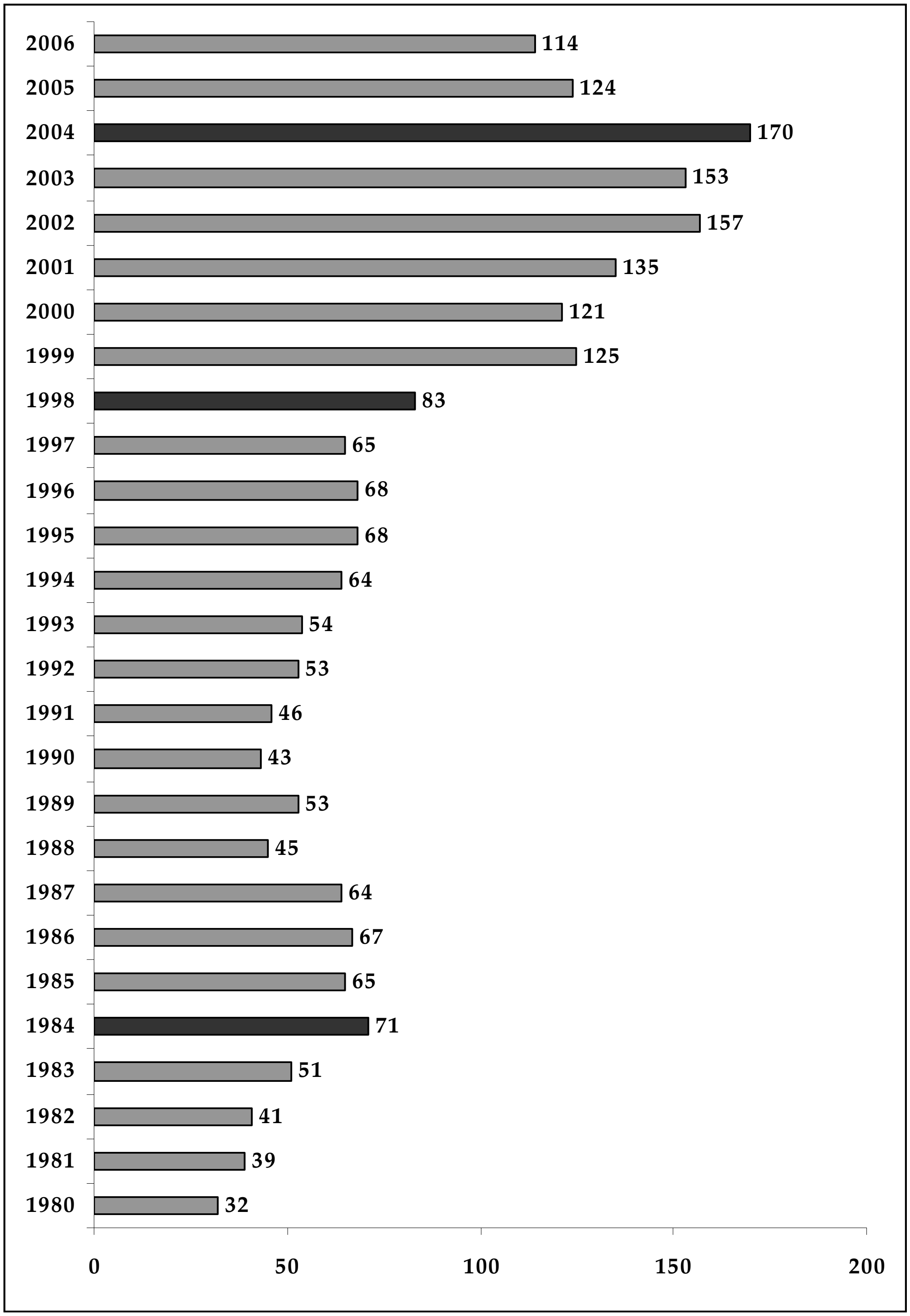
Even though the number of degrees has been falling since 2004 (and will continue to decline for the next couple of years), it is still 75% above the 1997 level.

TABLE 2.3: All degrees awarded with CSc designation from 1980 to 2006

Year	Number of CSc Degrees
1980	32
1981	39
1982	41
1983	51
1984	71
1985	65
1986	67
1987	64
1988	45
1989	53
1990	43
1991	46
1992	53
1993	54
1994	64
1995	68
1996	68
1997	65
1998	83
1999	125
2000	121
2001	135
2002	157
2003	153
2004	170
2005	124
2006	114

FIGURE 2.5. All Degrees Awarded with CSc Designation

Note: This bar chart is plotted from data in Table 2.3



Another view is given in Table 2.4, which only includes students who have declared a BSc degree in the Faculty of Engineering, so it excludes all of the Combined Degrees offered by Computer Science. Historical numbers for these latter programs have proved very difficult to obtain. However, in the current year, there are 50 students in these programs. Table 2.4 also excludes all of the BSENG students, yet Computer Science is responsible for delivering about 70% of this program. There are 79 students in the BSENG program in the current year.

TABLE 2.4: Headcount of BSc (declared) in the Faculty of Engineering

Year	# BSc declared	Year	# BSc declared
1991/92	199	1999/00	677
1992/93	242	2000/01	768
1993/94	283	2001/02	813
1994/95	305	2002/03	1026
1995/96	357	2003/04	919
1996/97	364	2004/05	748
1997/98	407	2005/06	568
1998/99	524	2006/07	428

It is difficult to compare directly the numbers from the peak years with the last few years since, if they wished to be able to get into CSc courses, students had to declare their program as early as possible. With the enrolment pressure having ameliorated, this is no longer necessary and some students now wait until 4th year to declare their programs.

Table 2.5 shows the number of students graduated since 1999 and the types of degrees that they obtained. The degrees with co-op designation amount on average to 46% for BSc Major and 71% for BSc Honours (with peaks to 100% for 1999, 2001, 2002).

The same information can be viewed as charts to be able to detect trends more clearly. Figures 2.6 and 2.7 show the plots for BSc Major and BSc Honours degrees. For simplicity, the newer types of “Work Experience” designation available only since 2006 has been incorporated in the co-op set.

TABLE 2.5: Number of CSc degrees (Major and Honours only)

Note: These are degrees awarded at the Spring and Fall Convocations: co-op designation, work experience designation (W.E.) and others.

Year	BSc in CSc (Honours			BSc in CSc (Major)			Totals
	w/Co-op	W.E	Honours	w/Co-op	W.E	Major	
Jun 1999	0	–	0	38	–	22	60
Nov 1999	2	–	0	10	–	14	26
Jun 2000	2	–	1	37	–	48	88
Nov 2000	1	--	0	11	–	21	33
Jun 2001	1	--	0	34	–	48	83
Nov 2001	1	--	0	17	–	19	37
Jun 2002	2	–	0	48	–	44	94
Nov 2002	0	--	0	23	–	26	49
Jun 2003	1	–	3	43	–	53	100
Nov 2003	1	--	0	18	–	37	56
Jun 2004	3	--	1	39	–	46	89
Nov 2004	3	–	2	21	–	37	63
Jun 2005	0	–	3	41	–	53	97
Nov 2005	2	–	2	13	–	28	45
Nov 2006	2	0	1	15	4	13	35
Jun 2006	3	0	3	27	4	34	71

FIGURE 2.6. BSc Major degrees from 1999 to 2006 (plotted from Table 2.5)

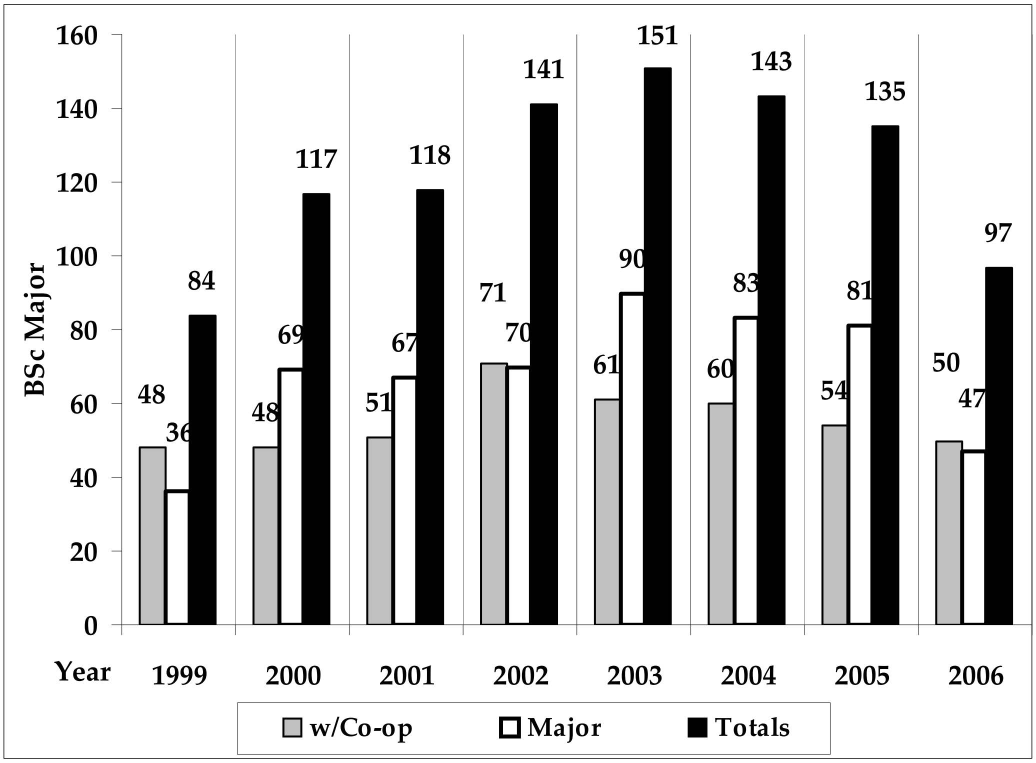


FIGURE 2.7. BSc Honours degrees from 1999 to 2006 (plotted from Table 2.5)

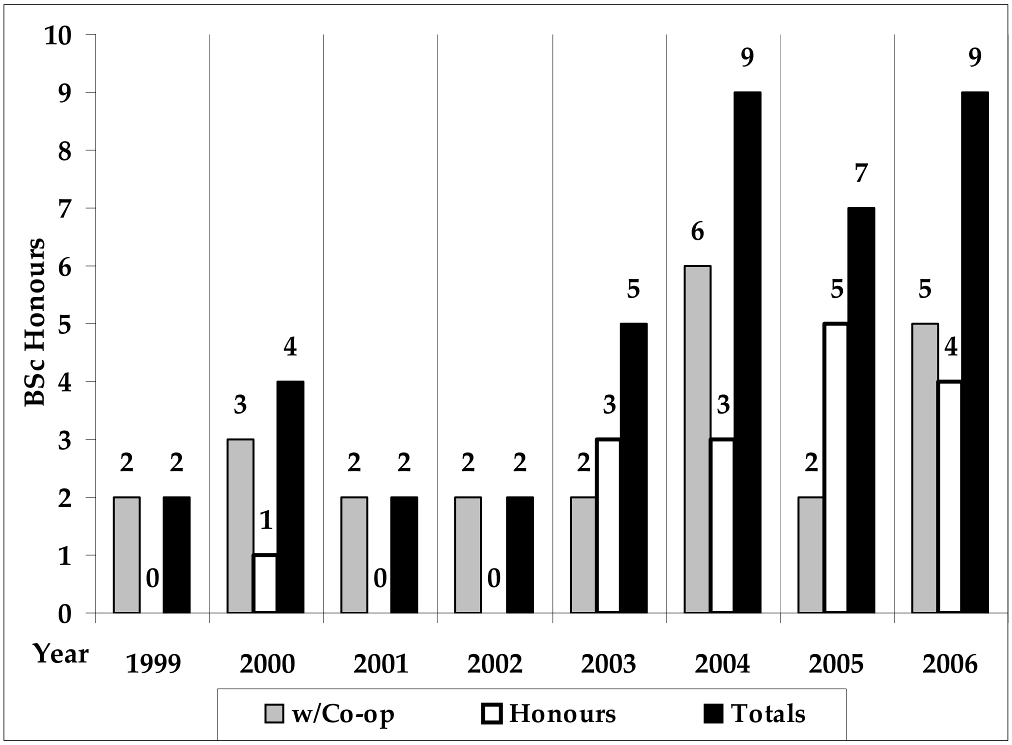
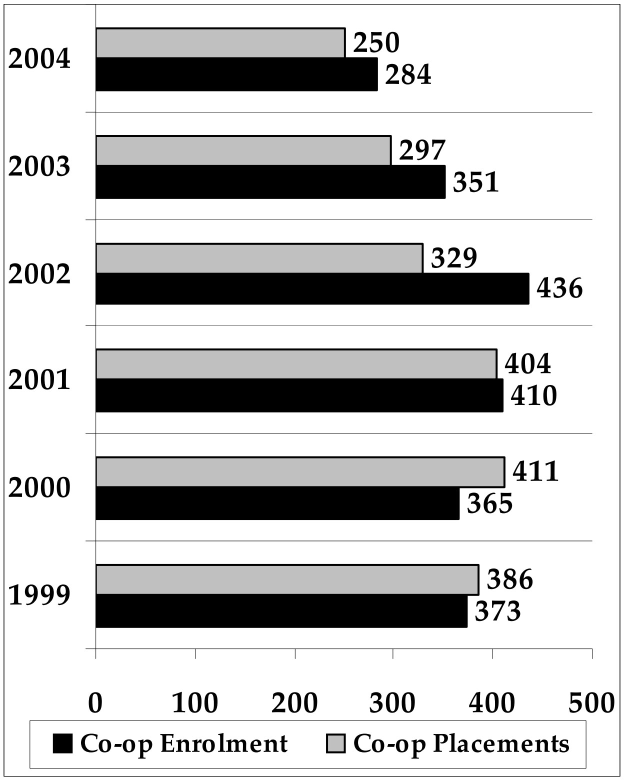


Figure 2.8 shows the data in Table 2.6 and its chart for the annual total undergraduate enrollment (winter term) and number of undergraduate work term placements (for the period September – August) of the Computer Science/Math Co-op Program from 1999 to 2004. These figures were obtained from the central co-op office, and are based only on the numbers of co-op students who actually received work term placements (as opposed to those registered in the program). Students in all undergraduate degree programs are included (CSc Major, CSc Honours, combined programs, Mathematics and Statistics)

FIGURE 2.8. Computer Science/Math Co-op: enrollment and placements

TABLE 2.6: Co-op Program: enrollment (winter term) and work term placements

Year	Co-op Enrolment	Co-op Placements
1999	373	386
2000	365	411
2001	410	404
2002	436	329
2003	351	297
2004	284	250



Similarly, Figure 2.9 shows the data in Table 2.7 and its chart for the annual total undergraduate enrollment (for each term) by degree program, and the number of undergraduate work term placements. This includes all students registered in the co-op program; that is, it is the number reviewed on each term by the co-op Retention Committee.

FIGURE 2.9. Enrollment (for each term) by degree program

Note: These charts include the number of work term placements.

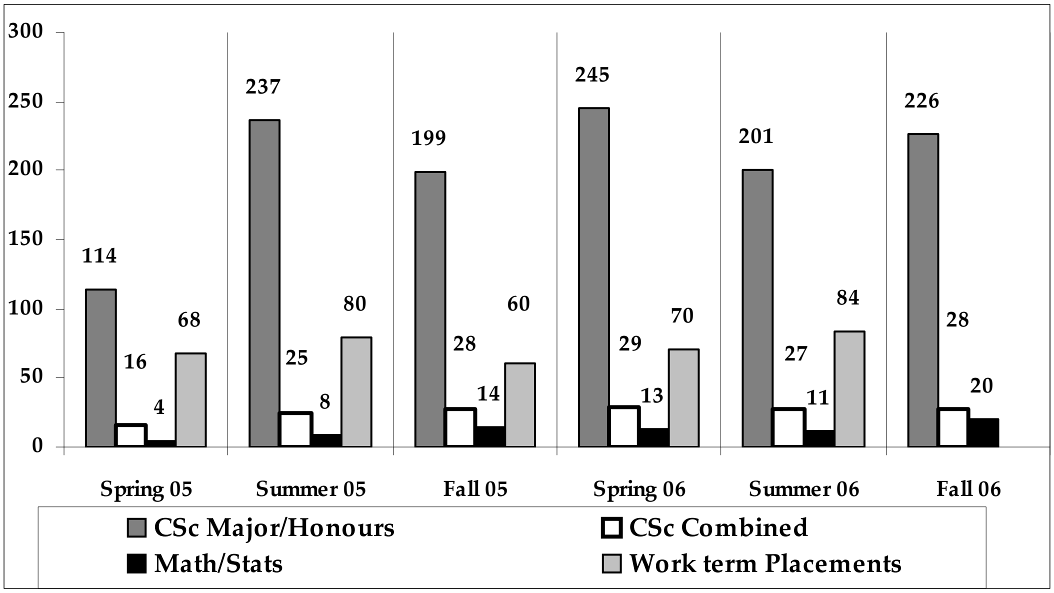


TABLE 2.7: Computer Science/Math Co-op Program Enrollment

Note: These are enrollments (for each term) by degree program and include work term placements

Term	Number of Students by Degree Program			Work Term Placements
	CSc Major/ Honours	CSc Combined	Math/Stats	
Spring 2005	114	16	4	68
Summer 2005	237	25	8	80
Fall 2005	199	28	14	60
Spring 2006	245	29	13	70
Summer 2006	201	27	11	84
Fall 2006	226	28	20	

As mentioned in the overview to this section, our students have had much success in acquiring jobs, both within the Co-operative Education program and after graduating. Recent ongoing co-op replacements have been obtained at Abebooks (e-commerce for booksellers), Electronics Arts, Genologics (life-sciences software), IBM Canada, New Heights Software (desktop telephony solutions), Research in Motion and Sierra Systems Group (IT consulting). Some graduates also go on to permanent employment at the companies just listed. Other companies who have hired our graduates include Apple, CitiBank, Google, Lucent, Microsoft, Nokia and Oracle.

2.11 Teaching Quality

The quality of teaching within the Department of Computer Science is outstanding, as is evidenced by higher than average teaching evaluation scores within the Faculty of Engineering. Two faculty members have received the Faculty of Engineering Award for Teaching Excellence (Dale Olesky in 2001 and Mary Sanseverino in 2003). Five of our graduate students have been honoured with the Andy Farquharson Award for Excellence in Graduate Student Teaching, which honours graduate students for their teaching of courses within the department: John Aycock (2000), Jesse Bingham (2001), Stirling Chow (2003), Chris Falk (2004) and Derek Church (2005).

Every term, in all courses, a student course evaluation survey is conducted (using a form that is common to the Faculty of Engineering). The statistical results of this evaluation are provided to the instructor and to the Chair and form part of the official teaching dossier of the faculty member. In addition the complete set of all student comments can be included in the teaching dossier.

The University requires that all faculty members maintain a teaching dossier that is used for the purposes of salary evaluation and all reappointment, promotion and tenure decisions. In addition, the department has a policy that any faculty member being considered for reappointment, promotion or tenure has peer teaching reviews conducted by three members of the department, chosen by mutual agreement between the faculty member and the relevant departmental committee.

In the last few years, Computer Science has had a very close relationship with the Learning and Teaching Centre (LTC) that is indicative of the importance that is attached to quality of teaching by members of the department. This is illustrated in the section below. Many members of the department have also attended course redesign workshops and other similar workshops organized by the LTC, and this has been very beneficial in helping the department to both improve the student experience in our courses and to move towards more experiential learning. Such attendance is viewed very positively by departmental promotion and tenure committees.

2.11.1 STUDENT EVALUATIONS

There is, of course, a large volume of statistical data from the student evaluations of each course. Here we just include averages from all respondents for each year to the question, “Overall teaching ability is excellent”. The range for the answers is [-2, +2].

The available data supports the idea of the very high quality of teaching in the department.

TABLE 2.8: Evaluation for “Overall Teaching Ability”

	CSc		Faculty of Engineering	
	Number of respondents	Score	Number of respondents	Score
2001/02	3416	1.01	6276	0.95
2002/03	3633	0.92	6306	0.87
2003/04[a]	1735	1.19	3743	0.99
2004/05	2596	1.17	5666	0.95
2005/06	2470	1.05	5399	0.95

[a] The data for Fall 2003 was lost, so this is based on Summer 2003 and Spring 2004

2.11.2 LEARNING AND TEACHING

The Learning and Teaching Centre at UVic is a very dynamic organization with the mission:

> *“to enhance an optimum environment for learning by encouraging and promoting university policies that give recognition and resources for continuous teaching improvement for all those who teach at the University of Victoria.”*

In the current year, the following members of the department have official involvement with the LTC:

- Daniel German: Learning and Teaching Scholar, 2006–07. A part time secondment to the LTC.
- Lior Malka: LTC TA Associate, 2006–07.
- Jon Muzio: LTC Advisory Council.
- Mary Sanseverino: LTC Associate Director IT Development, 2004-07. This is a half time secondment from the department to the LTC.
- Margaret-Anne Storey: LTC Advisory Council.

Learning and Teaching Development Grants

The Learning and Teaching Centre awards, through a competition, *Development Grants*. These grants are normally given to assist in course or program redesign. Department recipients in 2002 – 07 have been:

- Janet Bavelas (Psychology) and Daniela Damian, Psychology, 2003-04: *Development and Evaluation of Peer Collaboration*.
- Daniela Damian and Elizabeth Hargreaves, 2004-05: *Internationalization of software development: Learning to develop software in geographically distributed, multi-cultural teams.*
- Daniel German, 2006-07: *Evaluating the use of c-spatia and Scale-up in CSC 370 Database Systems.*

- Daniel German, 2002-03: *Creation of Atila for Automatic Testing of Programming Assignments.*
- Daniel Hoffman, 2005-06: RouterLab: *Hands-on study of the internet infrastructure.*
- Ulrike Stege, 2006-07: *A template for interdisciplinary, modeling-based computer science courses.*
- Ulrike Stege, 2002-03: *Teaching Algorithms and Data Structures with Web Animations: Application to CSC 225 and Other Interdisciplinary Applications.*
- Margaret-Anne Storey, 2005-06: *Using technology to enhance collaborative and community oriented learning.*
- George Tzanetakis, 2005-06: *Bridging music and programming: Music information retrieval.*
- Jens Weber-Jahnke, 2006-07: *Problem-based Learning Patterns for Practical Computer Science and Software Engineering Courses.*

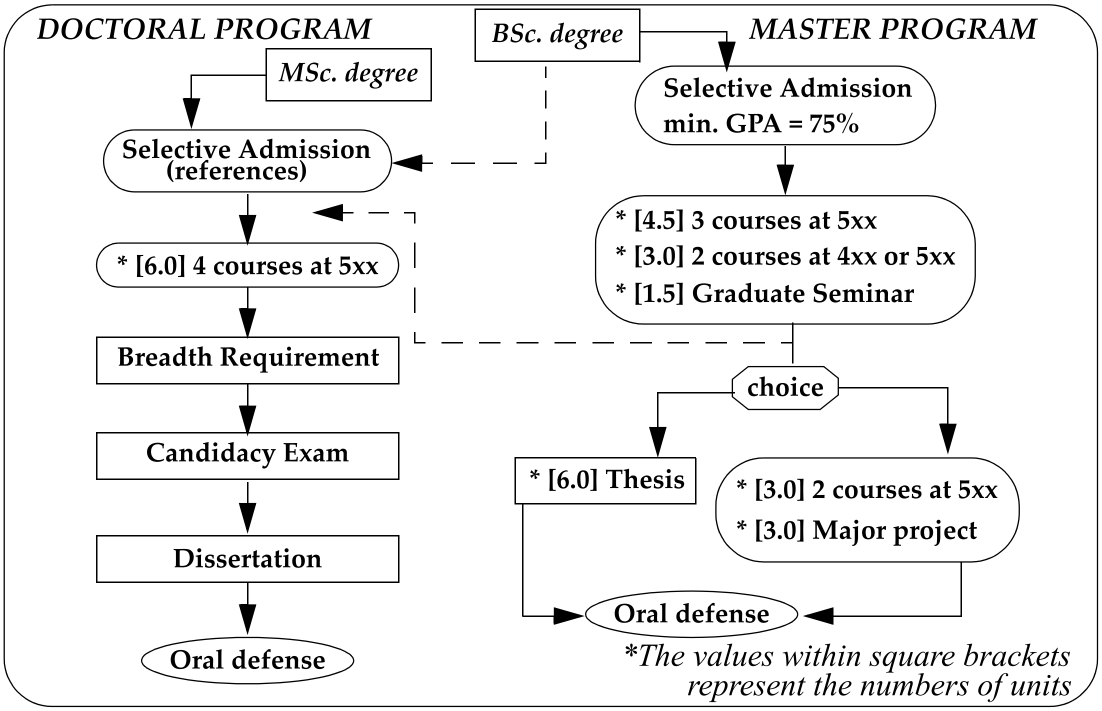
3: The Graduate Program

Taking a Byte Out of the Future is the motto found on the main web page of the Graduate Program. Our graduate students have an excellent record of employment in challenging jobs in academia, industry and government. In the near future, we believe that a graduate CSc degree will become an almost mandatory entry level qualification for significant careers that involve innovation and research. Our choices of paths towards an MSc degree guarantee that we fully support both the long term academically oriented student, and the industry-bound or even currently employed professional who needs to expand their knowledge of state-of-the-art theory and technology. Our PhD program maintains its excellent quality and depth, and continues to expand in scope and areas along with the growth in the number of faculty members. The department is expanding its graduate enrollment, as part of the overall trend in graduate students recruiting of the university.

3.1 Structure of the Programs

The department offers studies leading to the degree of Master of Science (MSc) and Doctor of Philosophy (PhD). The programs include a thesis or research project (MSc) or dissertation (PhD) as a core requirement, and both involve courses and a seminar course. The programs are summarized in the diagram of Figure 3.1. The dashed arrows denote possible paths which can be followed only with permission. More details are found in the section regarding admission and other policies.

FIGURE 3.1. The requirements of the Graduate programs.



3.2 Courses in the Curriculum

Many courses are offered to graduate students and it is transparent to them how they may be clustered from an administrative and workload view. However it is useful to view the organization of topics and courses from these other perspectives in order to consider both a vision and a strategic plan for the future. Not all courses are offered every year, yet a balance is usually attained among research areas.

3.2.1 REGULAR COURSES

Regular courses are listed in the calendar and have been established for some time. Great efforts have been made to keep courses current and, most of all, to introduce new courses, usually after they have been offered as “Topics” courses, as new faculty members join. The list of graduate courses offered from 1999 to 2006 can be found in Table 3.1. The shaded entries denote new courses, now formally in the calendar, introduced in that period of time.

TABLE 3.1: Courses for Graduate Students Offered from 1999 to 2006

Number	Title	Years
CSC 595	Graduate Seminar	1999-2006 (2)
CSC 405/505	Computer Graphics	2000, 2001, 2002, 2003, 2004, 2005, 2006
CSC 425/520	Analysis Of Algorithms	1999, 2000, 2002, 2003, 2004, 2005
CSC 426/526	Computational Geometry	1999, 2001, 2003, 2004, 2005, 2006
CSC 435/583A	Compiler Construction	2002, 2004, 2005, 2006
CSC 445/545	Operations Research I: Linear Programming	2000, 2001, 2002, 2003, 2004, 2005, 2006
CSC 446/546	Operations Research II: Simulation	2001, 2005, 2006
CSC 449/540	Numerical Linear Algebra	1999, 2000, 2002, 2004, 2006
CSC 450/550	Computer Communications and Networks	1999-2006 (3)
CSC 454/554	Fault Tolerant Computing	2000, 2001, 2002, 2003, 2004, 2005(2), 2006
CSC 460/560	Design And Analysis of Real Time Systems	2000, 2001, 2002, 2005, 2006
CSC 461/561	Multimedia Systems	2003, 2004, 2005, 2006
CSC 462/562	Distributed Computing	2001, 2002, 2004, 2006
CSC 524	Computational Complexity	2004, 2005
CSC 551/ SENG480A	Computer Communications and Networks II	2005, 2006
SENG 420/520	Software Evolution	2000 (2), 2002, 2003, 2004
SENG 450/550	Network Centric Computing	2000 (2), 2001, 2002
SENG 430/530	Object Oriented Design	2001, 2002, 2005

3.2.2 TOPICS COURSES

Similarly to the undergraduate program, topic courses are also taught and eventually they find their way into the calendar as a formal recurrent offering, after evaluation and curriculum discussion. Their list from 1999 to 2006 is in Table 3.2 and they are mostly new courses. The long list shows that, with the rapid growth in faculty members, the number of new topics courses has increased tremendously. One of the current priorities of the Graduate Studies Committee this year is to review and consolidate the course offerings and propose to the department which graduate courses should appear as regular part of the curriculum for the upcoming calendar.

TABLE 3.2: Topics Courses Offered from 1999 to 2006

Number	Title	Years/terms
CSC 464/586D	Concurrency	2005, 2006
CSC 480A/586A	Component Security	2003
CSC 480A/586A	Information Visualization and Knowledge Management	2003
CSC 482A/582	Advanced Cryptography	2006
CSC 482A/582	Algorithms for the Internet	2001
CSC 482A/582	Computer Game Design.	2000, 2001
CSC 482A/582	Combinatorial Algorithms	2001
CSC 482A/582	Cryptography	2000, 2001, 2003, 2004, 2006
CSC 482A/582	Graph Algorithms	2006
CSC 482A/582	Maple Flavored Concrete Mathematics	2002, 2005
CSC 482A/582	Practical Graph Algorithms	1999, 2003
CSC 482B/523	Randomized Algorithms	2000, 2004, 2005
CSC 482B/582B	Algorithms & Heuristics for NP-hard Problems	2003
CSC 482B/582B	Graph Algorithms for Chemistry	2005
CSC 482B/582B	The Art of Computer Programming: Volume 4 (Combinatorial Generation)	2004
CSC 482C/524	Computational Complexity	2002
CSC 482C/582C	Network Reliability	2005
CSC 483A/583B	Advanced Topics in Hypermedia	2003
CSC 484A/589A	Classical and Modern Methods of Optimization	2002
CSC 484A/ 589A/MUS 490/	Music Information Retrieval	2005
CSC 485A/586A	Planning and Implementing E-commerce	2002
CSC 485A/563/ SENG 480B	Data Compression	2003, 2004,2006
CSC 485A/585a	Testing and Reliability of Digital Systems	2001, 2004
CSC 485A/586	Multimedia Systems	2000, 2001
CSC 485A/586A	Wireless Mobile Networks	2003, 2004, 2005
CSC 485A/586A /SENG 480A	Visualization	2006
CSC 485A/ SENG480A	Implementation of Virtual Machines	2005
CSC 485B/581A	Logic and Artificial Intelligence	2002, 2005, 2006

TABLE 3.2: Topics Courses Offered from 1999 to 2006 (Continued)

Number	Title	Years/terms
CSC 485B/586B	Digital Documents	2000
CSC 485B/586B	Knowledge Discovery and Data Mining	2001
CSC 485B/586B	Network Firewalls	2004, 2005
CSC 485B586B/ SENG480B	Formalisms for Systems Design	2005
CSC 485B/587	Human Computer Interaction and eCommerce	2000 (2)
CSC 485C/552	Advanced Switching Theory	2004
CSC 485C/ 586C, now CSC 428	Computational Biology	2001, 2002, 2004, 2006
CSC 485C/586C	Multi-Agent Systems	2001, 2002(2)
CSC 485D/586D	Planning and Implementing E-Commerce	2000
CSC 485D/ 586D/SENG 480D	Database System Implementation	2006
CSC 576B	Global Software Development	2005
CSC 586Aa	Broadband Data Networks: Admission and Congestion Control Based on Knapsacks and Prices	2004
CSC 586C/ SENG480A	Web Services	2003, 2004
CSC 586E	Open Source Software Engineering	2005
CSC 586F	Advanced Topics in Human Computer Interaction	2006
CSC 586a/ SENG480A	Software Architecture	2000
CSC 586B/ SENG480A	Component Oriented Software Construction	2002, 2004
CSC 576A/ SENG480A	Topics in Software Development and Evolution	2001
CSC 586B/ SENG480B	Software Engineering Applied to World-Wide Web	2002

3.2.3 DIRECTED STUDIES COURSES

“Directed Studies” (CSC 591) courses are also taught, extra-to-load, mostly to individual students. This is sometimes inevitable if a particular topics course is not in the rotational plan for a couple of years and some students may be disadvantaged in their research. For each course, a formal outline must be provided and approved by the student and the instructor, plus the supervisor and the department Chair.

A few years ago, the Chair instituted a policy, both to encourage and reward appropriately graduate teaching at an advanced level. Any graduate course scheduled and then found to have an enrollment of less than 10 students is not necessarily cancelled, but is technically described as a group “Directed Studies”. A count is kept, such that a faculty member obtains teaching credit equivalent to one course after a total of ten (Graduate) Directed Studies. The practice has been well established and considered successful for all parties. This does not preclude at all that the objective is the continual increase in formal graduate level courses as a crucial ingredient for further expansions of the graduate program.

3.2.4 SPLIT NUMBERING/LEVEL IN COURSES

Many graduate courses are also co-listed with 400-level undergraduate ones. This has been necessary from a purely practical motivation to utilize optimally the resources in the department. It also provides an effective means to formally include graduate teaching effort within a teaching-load discussion at UVic, which traditionally emphasizes undergraduate teaching. The practice has been a topic of great discussion. While it is desirable to have classes for graduate students only, and it may indeed be a pedagogical advantage for both groups, often the numbers enrolled do not justify a separate course and this causes a great imbalance in faculty's teaching loads. Moreover, the higher administration tends to discourage class sizes less than 10 students (even if they are not explicitly forbidden as in other universities). Some of the advantages and disadvantages of such a practice are summarized in Table 3.3 for discussion. It must be emphasized that separate course outlines are provided and a clear difference must be shown between the expectations and workload of the two types of students, before approval by the Chair.

TABLE 3.3: Split numbering of 4XX/5XX level courses

Advantages	Disadvantages
• senior undergraduates are exposed to higher level project	• larger class sections
• senior undergraduates mingle and co-operate with graduate students	• less individual attention to graduate projects
• graduate students consolidate their basic knowledge or acquire it	• only a portion of the material can be at a truly advanced level
• intellectual resources and physical requirements (e.g. labs, software) are optimally utilized;	

3.2.5 GRADUATE SEMINAR

All graduate students must also attend the Graduate Seminar (CSC 595) at least once in their graduate career. While a few years ago this included mainly preparing students towards giving well prepared and fluent oral research presentations, the course has evolved. It is now offered twice a year and it normally involves:

- the preparation of a research proposal (similar to an NSERC grant application);
- the attendance to seminars from the science librarian for training in formal literature searches (not just Google!) using the latest databases available;
- the preparation of a complete bibliography on a research topic, together with the use of automated tools and integration into some appropriate future thesis platform;
- two technical oral presentations at the expected level of a conference session, using published articles from one's supervisor or a member of the research group (this has the added advantage of exposing all graduate students to the overall research of the whole department); and
- one-on-one feedback sessions on public speaking and technical writing.

3.3 Details of the MSc Program

It is important to emphasize some details of the MSc program, starting with sample projected plans of study.

Possible MSc Plan of Study (with Thesis)			
Term 1	• 2 courses + Seminar • TA work (max 12 hours/week)	OR	• 3 courses • TA work (max 12 hours/week)
Term 2	• 3 courses • TA work (max 12 hours/week)		• 2 courses + Seminar • TA work (max 12 hours/week)
Term 3	• Research and write thesis + TA work (max 12 hours/week)		
Term 4	• Research and write thesis + TA work (max 12 hours/week)		
Term 5	• Finish up + TA work (max 12 hours/week)		

For an MSc with a Project (instead of a Thesis), 2 more graduate courses need to be taken usually in Term 3.

In summary, an MSc degree requires five courses plus the Graduate Seminar and a Thesis. At most two of the five courses can be at the 4xx level, while the remaining must be at the graduate level. The option of an MSc degree with a Project requires seven courses plus the Graduate Seminar and a Project, with the same restrictions on the courses. The choice between the two types of programs has always existed and provides an excellent avenue especially for students who need the wider breadth of more graduate courses (great for people already in industry). The plans above expect a degree to be completed within 20 months.

3.4 MSc Committees: Supervisory and Oral Examination

The composition of the Supervisory Committee for a thesis MSc is formalized by the Faculty of Graduate Studies as shown in Table 3.4.

TABLE 3.4: MSc Committees

Program	Supervisory Committee	Defense Committee
MSc with Thesis	(1) Supervisor from CSc (2) Member from CSc (3) Member from CSc or from UVic (normally)	(1) *Supervisory Committee* (2) External Examiner from outside CSc (most often from UVic) (3) Defense Chair, not from CSc, appointed by Grad. Studies.
MSc with Project	(1) Supervisor from CSc (2) Member from CSc	(1) *Supervisory Committee* (2) Defense Chair from CSc

When a student is ready for a defense, the following process, prescribed by the university, must take place:

1. The student receives the approval on the thesis document from the main supervisor.
2. The student receives the approval on the thesis document from the other members of the supervisory committee. A reasonable frame of time (2-3 weeks) should be given for this task.
3. All members of the supervisory committee sign the "Request for Oral Examination" university form which states: *"The above signatures indicate that all committee members have examined the thesis/dissertation, and are satisfied that it represents an examinable document for the degree requirements, and also agree that they are available for the oral on the specified date and time."*
4. The Graduate Advisor signs the same form which states that: "*The signature of the Graduate Adviser indicates that correct administrative procedures have been followed within the Department or School. The signature also indicates that correct thesis/dissertation preparation procedures have been followed and that a copy of the thesis/dissertation is available in the Department/School General Office.*"
5. At the same time an external examiner for the oral defense must be appointed and named on this form, the choice being made normally by the supervisor and the student together, and a time and date for the oral defense agreed upon by all parties.
6. The "Request for Oral Examination" university form must be completed and forwarded to the Office of the Dean of Graduate Studies at least 20 working days before the anticipated date of the oral examination.

3.5 Issues for the MSc Program

Practices are well founded, and policies and procedures are developed either by the department or are university-wide. The main ones under current discussion are briefly summarized here, and they reflect the fundamental attitude that almost everything needs to be considered and evaluated in the context of time and changing expectations on a continuing basis, to maintain a growing successful program.

(a) *Breadth requirement.* As approved by the department as a whole, the calendar states that there is a breadth requirements for the courses in an MSc program. This stems from the view that, while an MSc is indeed the chance for a student to specialize in depth in a main area of interest, it would be inappropriate for all the courses in the program to be in the same area as the thesis topic. The exact implementation of the spirit of the requirement has not been formally approved yet, as a good policy must also be practical in its enforcement. At the moment the Graduate Advisor guides supervisors and students to follow rules similar to those stated by the University of Waterloo, namely that "*no more than three of the courses should be in a single area*". The areas are the same as used for the breadth requirement of the PhD program.

(b) *Oral defense.* The procedures for an oral examination of the thesis is seen by some as rather cumbersome, especially compared to other Canadian departments. However, they are set by the university as a whole and it is probably difficult to modify them.

(c) *Time to completion*. The average time to complete a program can be computed as 33 months (2 years and 9 months) for the past six years, but the number is misleading. Degrees with co-op involve at a minimum an extra 8 months of work-terms (often 12 or 16). Moreover, even the people who take longer cannot be seen as "career" graduate students, as they often have a full time job and continue writing their thesis and meeting with the supervisor on a part-time basis only. A near-future plan involves looking at the programs with a fine grained lens, evaluating the lengths of times, and decide whether there might be any structural changes to be made to support our students towards finishing earlier.

3.6 Details of the PhD Program

The details of the PhD program are described in the formally approved PhD Regulations. The main steps towards the completion of a PhD are summarized here:

Courses	Four graduate level courses (5xx).
Breadth	The Department of Computer Science believes that any candidate for a PhD degree, before graduating, must show a firm grasp of the overall field of Computer Science. The PhD Breadth requirement ensures that this goal is fulfilled, normally by taking a advanced courses in a broad range of categories and areas. The process to evaluate the breadth requirement can be found in below.
Candidacy	The candidacy examination is a requirement of the Faculty of Graduate Studies and cannot be waived by any department. However, the precise form, content, and administration of such examinations are determined by individual departments. The process implemented here can be found below. The UVic regulations state that: "*Within two years of registration as a provisional doctoral student and at least six months before the final oral examination, a student must pass a candidacy examination. The purpose of the candidacy examination is to test the student's understanding of material considered essential to completion of a PhD and/or the student's competence to do research that will culminate in the PhD dissertation.*"[a]
Dissertation	The standards for the dissertation are set by the Faculty of Graduate Studies which states: "*The doctoral dissertation must embody original work and constitute a significant contribution to knowledge in the candidate's field of study. It should contain evidence of broad knowledge of the relevant literature, and should demonstrate a critical understanding of the works of scholars closely related to the subject of the dissertation. Material embodied in the dissertation should, in the opinion of scholars in the field, merit publication.*"[a]

[a] Graduate Calendar.

3.7 PhD Committees: Supervisory and Oral Examination

The composition of the Supervisory Committee is formalized by the Faculty of Graduate Studies as shown in Table 3.5.

TABLE 3.5: PhD Committees

Supervisory Committee	Defense Committee
1. Supervisor from CSc **2.** Member from CSc **3.** Member from CSc **4.** Outside member: from UVic, not in CSc	**1.** *Supervisory Committee* **2.** External Examiner from outside UVic **3.** Defense Chair, not from CSc, appointed by Grad. Studies.

When a student is ready for a defense, the process is similar to the one for an oral defense for an MSc thesis, with the following differences:

1. The waiting time between submitting the "Request for Oral Examination" form and the anticipated defense date is lengthened to 30 days.
2. The external examiner must be approved by the Dean of Graduate Studies after submission of a CV, must be considered an expert on the subject and not in any conflict of interest.
3. The external examiner must submit a written report on the thesis work to the Dean of Graduate Studies at least 2 weeks before the anticipated defense date.

3.8 Issues for the PhD Program

The PhD Regulations are in the process of being updated and revised, where necessary. While the overall program is expected to remain the same, as far as the formal requirements, the main focus is to define more accurately the detailed departmental procedures for the two milestones, namely the candidacy examination and the breadth requirement.

The average time to complete a program can be computed as 52 months (4 years and 4 months). Similarly to the MSc program though, a more fine grain evaluation is about to be done.

3.9 Recruitment, Admissions, Awards

The normal expectation for entry into a Masters program is a BSc degree in Computer Science. For a Doctoral program the expectation is to have an MSc degree in Computer Science. Other degrees are accepted as equivalent when the fields are appropriate, especially towards a more interdisciplinary graduate program, possibly with extra courses required. Table 3.6 and 3.7 summarize the admission requirements for both programs. Over the last 7 years the department has received and evaluated between 70 and 120 applications per year and has admitted between 15 and 35 new students per year.

TABLE 3.6: Admission to the MSc Degree

Prerequisites	Must satisfy the UVic admission requirements.
Previous Degrees	Major or honors undergraduate degree in Computer Science/Computer Engineering/Software Engineering (or its equivalent); *OR* a major or honors degree in Mathematics with an emphasis on Computer Science.
Grades	A minimum of B+ (75-79%) is required for courses taken in the last two years of the BSc
GRE	The GRE test is not required but it is highly recommended for international students.
Language	Applicants whose first language is not English must have a minimum TOEFL score of 575 (paper test) or 233 (computer-based test).
Mature Students	Mature applicants who do not meet the minimum GPA of 5.00 (56%) may be allowed admission as a "Conditionally Admitted Mature Student" (C.A.M.S.) if they satisfy the following requirements: • at least 4 years have passed since the completion of a BSc degree; • the applicant has at least 4 years of relevant professional experience; • a recommendation by the department; • submission of a complete resume • approval by the Dean of Graduate Studies.
Exceptional Cases	A student who does not have a degree as stated above may be admitted to the program in rare instances, but normally will be required to complete additional make-up courses. In doing so, the student must obtain a grade of at least B (70-74%) in each make-up course, and an average of B+ (75-79%) overall in the make-up courses. Evidence of ability to pursue graduate studies must also be provided.
Decisions	Final decisions on admissions are made by potential supervisors.

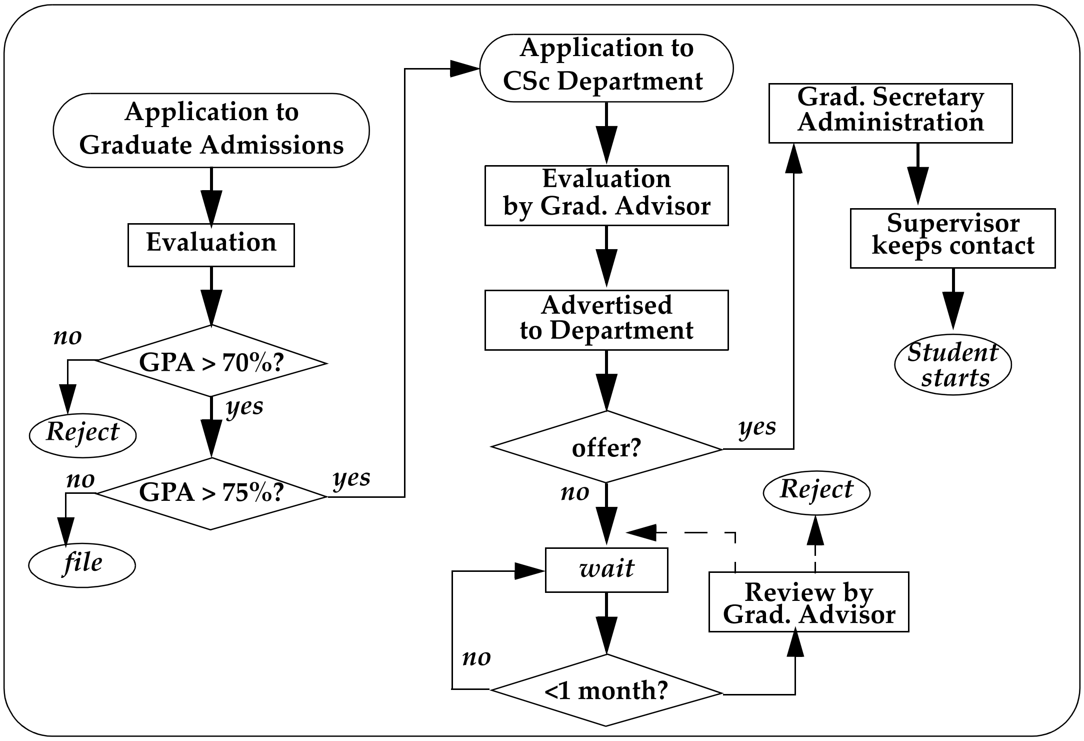
TABLE 3.7: Admission to the PhD Degree

Prerequisites	• Must satisfy the UVic admission requirements. • Must provide evidence of ability to undertake substantial original research. This typically includes letters of recommendation from qualified referees, the completion of an MSc thesis and scholarly work.
Previous Degrees	An MSc in CSc, or the equivalent, with a first-class standing.
Grades	First class standing (>79%) is required for courses taken in the last two years of the previous degree.
GRE	The GRE test is not required but it is highly recommended for international students.
Language	Applicants whose first language is not English must have a minimum TOEFL score of 575 (paper test) or 233 (computer-based test).
Mature Students	Mature applicants who do not meet the minimum GPA of 5.00 (56%) may be allowed admission as a "Conditionally Admitted Mature Student" (C.A.M.S.) if they satisfy the following requirements: • at least 4 years have passed since the completion of an MSc degree; • the applicant has at least 4 years of relevant professional experience; • a recommendation by the department; • submission of a complete resume; • approval by the Dean of Graduate Studies.
Exceptional Cases	If a student shows outstanding academic promise, he or she may directly enter the PhD program with only an undergraduate degree and will be required to complete additional make-up courses. The special application must be also approved by the Graduate Studies Committee.
Decisions	Final decisions on admissions are made by potential supervisors.

3.10 The Overall Process of Admission

Figure 3.2 shows the process followed by all admissions. The first assessment of an application is done by Graduate Admissions and Records. Once an application is complete and it meets the requirements of both the Faculty of Graduate Studies (GPA > B = 5 = 70-74%) and the Department of Computer Science (GPA > B+ = 6 = 75-79%), it is forwarded to the department for consideration. At this point the Graduate Advisor gives an initial evaluation with comments to each application and the highlights are summarized in lists circulated electronically. Each faculty member can examine all applications and makes an independent decision of admission based on a number of criteria, including: GPA, previous experience, area of research interest, suitability to current projects, references, availability of positions in a research group. Many faculty members also interview candidates. A good matching of supervisors and students and areas of research is done before any admission, such that *a student is sent a formal offer of admission if and only if a suitable supervisor has been found*. The department does not admit a fixed number of students every year, as the decisions are left to individual potential supervisors.

FIGURE 3.2. The admission process



The admission process is fairly labour intensive for everybody. The goal is to make a match between a faculty member and a prospective student before any offer of admission. While a student is free to change their supervisor later in the program, most associations have been successful, as the initial screening, with a possible interview, has resulted in careful decisions by both parties. This is different from other departments where a certain quota of students, variously distributed among areas, are admitted every year, possibly by a Graduate Committee using more general criteria. In this case students are expected to find and choose a supervisor within the first 2 terms of their program, while they are taking courses.

As the size of the department has grown, the admission process has been under discussion, towards the objectives of making it as effective and efficient as possible, maximizing recruiting, while competing with other universities. There are advantages and disadvantages to our process, summarized below.

Admission Process	
Possible Advantages	**Possible Disadvantages**
The careful matching of applicant and supervisor done up front has been very successful.	The matching is done from paper qualifications and possible interview only (unless previously a local student).
Students do not have to shop around for a supervisor while taking coursework.	There is less flexibility in the process as individual faculty members are committed to their own individual admissions.
Supervisors do not have to shop around among newly admitted students taking coursework.	
Financial arrangements are guaranteed in advance when an admission offer is made.	Some years some good students may not be admitted in an area because a matching supervisor with available funds is not available.
There is more flexibility in the process as individual faculty members are free to make their own admissions, without being guided by the choices of a departmental committee or other criteria based on quotas.	The process is quite labour intensive for everybody who wants to have graduate students.
The students have freedom to change supervisors, yet they are directly supervised and guided from the very beginning, jump starting on research projects and focusing earlier.	Some students may find it psychologically difficult to change supervisors, if they discover any disappointment in the work environment.

3.11 Financial Support

The department believes that adequate financial support of graduate students is a crucial factor in contributing to their overall success. For this reason, graduate students are normally admitted only if they can be guaranteed support during their studies. This support comes from a combination of four main sources:

1. scholarships
2. research grants
3. teaching assistantships
4. salaries paid by employers of part-time or from co-op workterms.

As a brief summary most students have financial support as the sum of:

- a minimum of 240 hours per year of Teaching Assistantship (TA) work for the department;
- research salary from the supervisor or from a scholarship, for an average of 40% of working time (Research Assistantship/RA); and
- supplement directly from Graduate Studies pro-rated according to workload (AIS).

The detailed amounts are shown in Table 3.8, Table 3.9 and Table 3.10.

The Research Assistantship (RA). This may come from: (a) an NSERC award, or (b) a University of Victoria Fellowship, or (c) research funds of the supervisor (labelled as "Standard Support"). Incoming students who hold NSERC awards are also normally awarded the President's Research Scholarship in the amount of $4000 for each year the award is held. UVic Fellowship holders normally receive a top-up award from the supervisor's funds of at least $4,300 per year. The UVic Fellowship is awarded for 1 year to MSc students and for 2 years to PhD students.

The Teaching Assistantship (TA). A student is expected to work as a teaching assistant for the department, either as a marker or as a lab instructor. An offer of admission normally guarantees a minimum of 240 TA hours per year, distributed over the three terms, but the workload never exceeds 12 hours per week for 10 weeks in a single academic term. The rates for TA work are set by the union (CUPE) contract with the university. UVic Fellowship holders and NSERC scholars may also apply for up to 120 hours of TA employment for up to two years of a Master's program, or for up to 4 years of a PhD program.

Academic Income Supplement (AIS). The Faculty of Graduate Studies provides the "academic income supplement" (AIS) which is administered by the department. AIS is pro-rated based on the number of Teaching Assistantship (TA) hours of work, provided one is a registered full-time student in the term in which the AIS is assigned. TA work and AIS payments are available only to full-time students who are on campus.

3.11.1 THE BOTTOM LINE

Tables 3.8, 3.9 and 3.10 summarize the financial support for the current academic year 2006-2007, for the three main categories.

TABLE 3.8: Standard Support (from September 2006)

	Research Assistantship (RA)		Teaching Assistantship (TA)	AIS	Total
MSc	$12,640	domestic	$5,268	$1,212	$19,120
	$13,238	international	$5,268	$1,212	$19,718
PhD	$13,767	domestic	$5,268	$1,212	$20,247
	$14,415	international	$5,268	$1,212	$20,895

TABLE 3.9: UVic Fellowship Support (from September 2006)

	UVic Fellowship	Top-U	Teaching Assistantship (TA)	AIS	Total
MSc	$13,500	$4,300	$2,634	$1,212	$21,040
PhD	$15,000	$4,300	$2,634	$1,212	$22,540

TABLE 3.10: NSERC Support (from September 2006)

	NSERC Fellowship	Top-U	Teaching Assistantship (TA)	AIS	Total
MSc	$17,300	$4,000	$2,634	$1,212	$24,540
PhD	$17,500	$4,300	$2,634	$1,212	$24,740

The financial support over the last 7 years is shown in Table 3.11. The centre column lists the AIS contribution from Graduate Studies which are the same for both programs. The amounts are included in the totals

TABLE 3.11: Graduate financial support from 2001 to 2007

Year	Total MSc	RA	TA	GTRF/ GTF/ GTAF/AIS	Total PhD	RA	TA
2001/2002	$13,891	$9,600	$4,291		$14,995	$10,704	$4,291
2002/2003	$15,491	$9,600	$4,291	$1,600	$16,595	$10,704	$4,291
2003/2004	$17,237	$12,080	$4,197	$960	$18,341	$13,184	$4,197
2004/2005	$18,054	$12,380	$4,197	$1,477	$19,158	$13,484	$4,197
2005/2006	$18,339	$12,640	$4,197	$1,502	$19,466	$13,767	$4,197
2006/2007	$19,120	$12,640	$5,268	$1,212	$20,247	$13,767	$5,268

It is useful to examine more closely the amounts, especially with a view to observe the provenance of the funds as a percentage of the total and of the yearly increases, as shown in Tables 3.12 and 3.13.

TABLE 3.12: RA and TA support as percentage of total support for MSc

Years	Increase Total Support	Increase R.A.	Increase of (TA+ AIS[a])	Ratio of R.A. to Total	Ratio of (T.A.+ AIS[a]) to Total
2001/2002				71%	29%
2002/2003	11%	0%		65%	35%
2003/2004	11%	23%	-12%	72%	28%
2004/2005	4%	2%	10%	70%	30%
2005/2006	2%	2%	0%	71%	29%
2006/2007	4%	0%	14%	68%	32%

[a] AIS here denotes contributions from Graduate Faculty, even when labelled over the years as GTF, GTAF, GTRF, AIS

TABLE 3.13: RA and TA support as percentage of total support for PhD

Year	Increase Total Support	Increase R.A.	Increase of (TA+ AIS[a])	Ratio of R.A. to Total	Ratio of (T.A.+ AIS[a]) to Total
2001/2002	-	-	-	69%	31%
2002/2003	12%	0%	-	62%	38%
2003/2004	11%	26%	-12%	70%	30%
2004/2005	5%	2%	10%	69%	31%
2005/2006	2%	2%	0%	69%	31%
2006/2007	4%	0%	14%	66%	34%

[a] AIS here denotes contributions from Graduate Faculty, even when labelled over the years as GTF, GTAF, GTRF, AIS

As can be seen from these tables, the total amount of financial support has been increasing over the years, keeping pace with inflation, if not tuition costs. The relative proportion between the university contribution, via the T.A. salaries and the AIS supplements, and the contributions via the R.A. funds from supervisors is somewhat skewed. The latter are the major part of the financial support for CSc graduate students and has remained at an average of 67 or 69% (for MSc and PhD respectively). It should be noted that in the year 2003/2004 the R.A. portion had to be increased considerably (by 23% and 26% for MSc and PhD respectively), since university-wide cutbacks lowered the availability of other funds by 12%.

3.12 The Budget from Graduate Studies

A budget is given every year from the Dean of Graduate Studies to the department. The total amount includes funds for UVic Fellowships and for “supplements”, which have been variously labelled over the years as “GTF”, GTAF, GTRF” and are currently called “AIS” (Academic Income Supplement). The budget is given to be administered locally by each unit, which can decide which portion should be given towards Fellowships and which portion towards other supplements, with a minimum guaranteed amount stated by the Dean for the latter.

The philosophy is that Fellowships are given for academic excellence, and are particularly useful to international students who are not able to compete for some Canadian scholarships. Supervisors are encouraged to give a top-up amount, as shown in Table 3.9. The supplements from the “AIS” are seen as an extra bonus to graduate student who already receive some form of support from the university as casual employees. Only those who are working and paid from a university account, and not from a research account, are eligible. Since almost all the CSc graduate students are also employed as TAs, the supplement is given pro-rated to the TA hours and is calculated every term. The budget figures for the last few years are in Table 3.14, divided between Fellowships and the other supplements, together with some statistics.

The current amounts stated for UVic Fellowships, as shown in Table 3.9, is $13,500 for MSc students and $15,000 for PhD The former receive the Fellowships for one year, while the latter are normally renewed for a second year, based on continuing successful performance.

TABLE 3.14: Contributions from Graduate Studies

Years	Total Budget	AIS/GTF spent	Fellowships	Increase in Total Budget	Increase in Fellowships
2004/2005	$187,964	$76,137	$111,827		
2005/2006	$218,579	$75,659	$142,920	16%	28%
2006/2007	$218,580	$75,580	$143,000	0%	0%

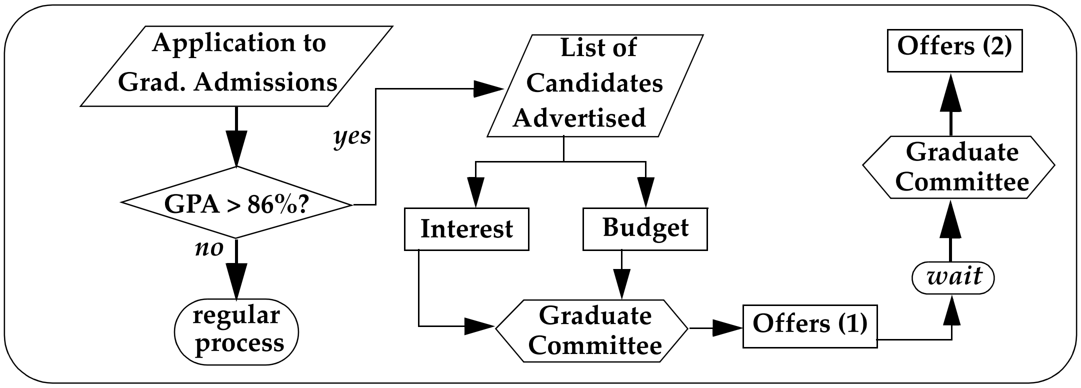
3.12.1 TUITION FEES

The fees currently set by the University are $1,527.40 [$1,817.50] for domestic [international] students per term. A student is expected to register every term (thus for 3 terms per year). An MSc degree is charged the full tuition fee for 5 terms, after which reregistration fees apply. For a PhD program, reregistration fees apply after 7.5 terms (approximately 2.5 years).

3.13 The Selection of UVic Fellowships at Admission Time

The process of selecting UVic Fellowship holders is summarized in the diagram of Figure 3.3.

FIGURE 3.3. The process of selection for UVic Fellowships



Fellowships are seen as a great bonus towards recruiting academically excellent students and they are very competitive. The process outlined in Figure 3.3 has been devised to ensure fairness and equity of distribution.

The decisions are made by the CSc Graduate Studies Committee. Applicants are selected based on:

- GPA
- References
- Publications (if any)
- Statement of interest and support from a potential supervisor(s).

The last clause is necessary since any graduate student is only admitted when matched to a supervisor from the start. It happens, as a side effect, that some excellent applicants with high GPA may not be selected, as no supervisor is interested and available. The minimum GPA internally is normally 86%, but very few Fellowships over the years have been awarded to students with less than 90% GPA.

3.13.1 ISSUES FOR FELLOWSHIPS

The main issue for the awarding of a Fellowship is the freedom to have entry points in the program other than the Fall of each year. The University still views the whole process based on the traditional assumption that academic graduate programs start in September. This department is more flexible and has the choice of starting also in January, for the Spring term, and in May, for the Summer term.

It is difficult at times to be completely inclusive of such students within the Fellowship process. On one hand, it is important to give the same opportunities to all. On the other hand, if money is kept aside for Fellowships to be awarded in January or May and no qualified applicants are available, the unused funds are returned to Graduate Studies. Yet the funds may have been useful in previous terms.

The flexibility of carrying over a small percentage of funds between budget years would be very helpful.

3.13.2 ISSUES FOR FINANCIAL SUPPORT

While it may be trite to state that more money is needed to support more students, it is important to emphasize that government direct financial support has been historically geared towards undergraduate studies, and more emphasis on graduate education and research must be undertaken.

At the moment, graduate students can be seen to be supported mainly by the individual efforts of faculty members in raising funds through their grants. The availability of funds, of course, affects recruiting directly.

3.14 Factsheet: Data on Graduate Students

There are many statistics and data which are very useful to view the program over the years. The current snapshot of enrollment is shown in Table 3.15.

TABLE 3.15: A Snapshot in February 2007

Total	Program		Gender	
	PhD	MSc	M	F
123	46	77	94	29
	38%	*62%*	*76%*	*24%*

The most important data are given by the number of degrees finished by year, as shown in Table 3.16 and in the chart of Figure 3.4.

TABLE 3.16: Graduate Degrees from 2000 to 2006

	M. Sc.	Ph. D.	Total
1999/00	12	3	15
2000/01	10	4	14
2001/02	14	2	16
2002/03	15	2	17
2003/04	32	5	37
2004/05	23	1	24
2005/06	14	1	15
2006/07	28	5	33
Totals:			*171*

FIGURE 3.4. Graduate degrees awarded from 2000 to 2006

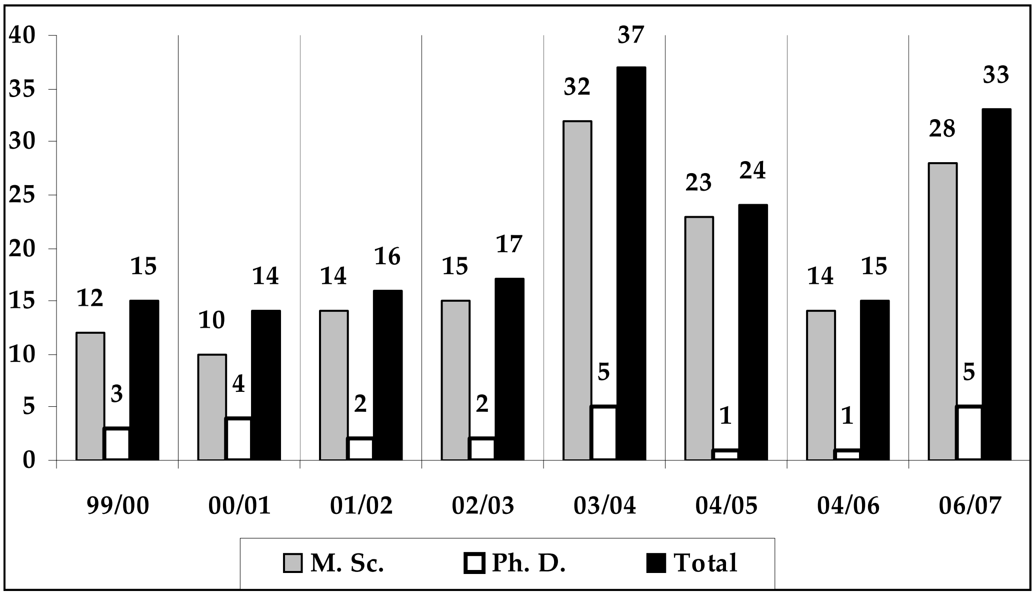


Table 3.17 shows the corresponding number of graduate students enrolled and some attributes.

TABLE 3.17: Graduate enrollment

	Gender		Study Habits		
	Male	Female	F/T	P/T	Total
1999/00	52	13	57	8	65
2000/01	55	20	67	8	75
2001/02	68	21	80	9	89
2002/03	76	25	89	12	101
2003/04	84	30	102	12	114
2004/05	89	30	116	3	119
2005/06	89	30	113	6	119
2006/07	85	31	109	7	116

3.15 Graduate Co-op Program

The graduate co-op program was formally introduced 10 years ago and has been expanding ever since. The original vision was based on enhancing the graduate experience through industrial collaboration at the thesis research level. Since it is encouraged and often desirable for a faculty supervisor to consult or be in direct contact with development work in industry, this form of collaboration could be labelled “technology transfer on two legs”. By supporting the research thesis work of a graduate student, normally through a financial contribution as salary for workterms, the outcome is positive for all parties involved. Thus graduate co-op can be seen as a research collaboration vehicle with the added benefit of financial support going directly to the graduate student.

For an MSc degree to be labelled as co-op, two workterms must be completed, while three are required for a PhD program. Usually the workterms are done consecutively with the same employer to maximize the scope of the research project. Normally students are allowed to enter a graduate co-op program only after two academic terms.

Over the years this vision has changed somewhat. The demand for experienced professionals in the field has grown so much that the pool of undergraduate co-op students was insufficient. Many graduate students, especially international ones or younger graduates who have not had much previous professional experience, have much to gain by enrolling in co-op work terms even when the job description may not be related directly to the thesis. The work itself, usually at a quite high level, is an excellent vehicle for learning. Additionally, employers have been enthusiastic about hiring excellent students, since the graduate candidates have already been filtered via the high academic standards of graduate admission.

A side-effect which can be viewed as less positive is that a student is often so successful in the co-op job that a full time permanent job offer may be made by the employer. Then the intensive job of writing the thesis may have to be completed part-time while the student is off-campus and consequently taking much longer to complete. While this can be difficult for both the student and the supervisor, it is, on the other hand, also a very positive indicator for the success of the graduate co-op program.

The number of graduate degrees with co-op designation is listed in Table 3.18, together with an even more interesting statistic – the number of workterms in which students have registered over the years. In interpreting these numbers, it should be noted that many students who have had workterms recently have not graduated yet.

TABLE 3.18: Graduate degrees with co-op and workterms

	Total Degrees	Co-op Degrees	Number of workterms
1999/00	15		
2000/01	14		
2001/02	16		
2002/03	17	6	9
2003/04	37	6	17
2004/05	24	5	15
2005/06	15	8	21
2006/07	33	-	22

3.16 Administrative Infrastructure

The graduate programs are administered collaboratively by the following:

Graduate Advisor:
- A rotating position for a faculty member appointed by the Chair for 2 years.

Graduate Studies Committee:
- Composed of five faculty members and a graduate student representative, plus the Graduate Advisor as chair.

Graduate Secretary:
- Appointed by the department.

The Graduate Studies Committee meets regularly every month and makes decisions on all departmental matters regarding the graduate program. Policies and procedures, when new or revised, are submitted for final approval to the department as a whole. The Graduate Advisor works in close collaboration with the Faculty of Graduate Studies, and the Associate Dean in particular.

3.17 Graduate Student Involvement

Graduate students sit on various committees in the department and their involvement is actively encouraged. The graduate representative sits as a full voting member on the department's Graduate Studies Committee, which is chaired by the Graduate Advisor. The graduate representatives attend all GSS (Graduate Student Society) meetings, and are informed about issues relevant to graduate students at UVic.

The GSS distributes some funding for graduate student activities. We applied for the GSS Departmental Grant, and received $400 to support social activities for CSc graduate students.

3.18 Social Activities and Collegiality: the Students' Views

The main goal of the less academic portion of the graduate programs is to strengthen the connection between research groups through social and academic activities on a regular and informal basis. In addition, there have been soccer and volleyball games and similar events organized by the CSCU (Computer Science Course Union) to which graduate students are invited. The current graduate student representatives also organized a career workshop, and helped directly with the Faculty of Engineering Graduate Innovation Forum (GIF). Other social events are often organized within each research group and to which other students are invited.

Current successful activities:

- *The CSC Coffee House:* a scheduled monthly mixer for graduate students. Each month a different research group hosts the event. Members of the group present themselves and their research. The coffee house helps both domestic and international graduate students make connections with fellow graduate students and with faculty members. The main objective here is to increase the interaction between research groups, students and faculty members, thus creating a stronger and healthier research community. The budget for the coffee house is $100 a month, paid by the department.
- *The Coffee Club:* for graduate students and faculty members, in the main 6th floor lounge. The current organizer is a graduate student, David Sprague, who also maintains a list of graduate students interested in ad-hoc social activities (movies, eating out, etc.).
- *Soccer and other Sporting Events:* for graduate students and faculty members. At least twice a year there is an enthusiastic soccer match. The plan is to expand to run other sporting events such as roof top volleyball.

Planned activities:

- *General Meetings:* to be scheduled monthly, starting in January 2007. The objective is to exchange feedback about the graduate program in a semi-formal setting.
- *CSC Travel Grant:* the prize for a research presentation series for graduate students. The best presentation can win a $400 travel award, on top of the travel funds which can be applied for through the Faculty of Graduate Studies. The objective of this series is to encourage graduate students to present their research to the entire department, and to practice their presentation skills.
- *More Workshops and Panels.* There is a strong desire to have workshops for graduate students on various relevant topics: how to submit and get papers accepted, how to optimize the outcome of graduate school, how to structure and develop a successful thesis, and so on. Such events have occurred in an ad hoc manner during the last few years.

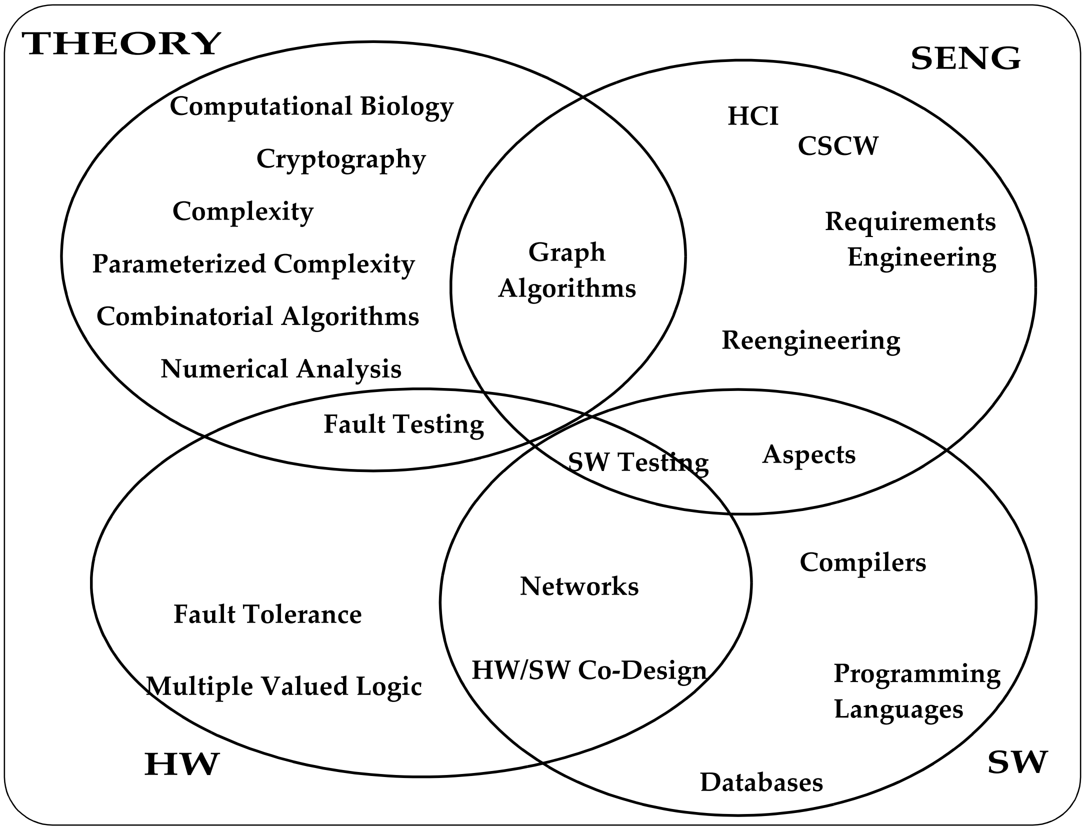
- *Professional Development.* Along the same lines of workshops, it is important to have a professional development series where industrial partners, such as IBM, can organize a talk or run a hands-on workshop for a given technology. Such events have occurred often and in an ad hoc manner over the last few years, especially when linked to the co-op program. However it is desirable to develop a more cohesive plan to ensure that every student can have some exposure at least a couple of times each year.

4: Research and Scholarship

The philosophy of the department from its earliest days has been to avoid spreading itself too thinly across all the possible research areas in computer science. The consequence is that not all research areas are represented. Only recently has the department hired new faculty whose primary research interest is in computer graphics, and it chose to do so only when at least two positions could be committed to the area. As it worked out, three appointments were made. The new appointments are described in more detail below (section 4.2). The main remaining gap in the department's current coverage of research areas is probably Artificial Intelligence (AI). Although some faculty have interests which overlap AI and give courses with an AI title, the department cannot be described as having an AI research presence.

It is hard to classify the diverse research areas of the department into a few broad categories. However Figure 4.1 attempts to provide just such a classification.

FIGURE 4.1. Major Research Areas in the Department (July 2006)



4.1 Faculty Research Interests

Table 4.1 lists all faculty members in the Department of Computer Science as of July 1, 2006 who are members of the Faculty of Graduate Studies, and are therefore permitted to supervise graduate students.

TABLE 4.1: Primary Research Areas of Research Faculty

Faculty Member	Primary Research Interests	Area
Mantis Cheng	Real-time and embedded systems, mechatronics	SW
Yvonne Coady	Aspect-oriented programming	SW
Daniela Damian	CSCW, requirements engineering	SENG
Sudhakar Ganti	High-performance networks	HW
Daniel German	Open source, software evolution	SENG
Daniel Hoffman	Software testing, networks	SENG
Nigel Horspool	Compilers, programming languages	SW
Bruce Kapron	Computational complexity, cryptography	THE
Valerie King	Randomized algorithms, combinatorial algorithms	THE
Michael Miller	Fault testing, CAD, multiple-valued logic	HW
Hausi Müller	Software engineering, software evolution.	SENG
Jon Muzio	Fault tolerant computing, multiple-valued logic	HW
Wendy Myrvold	Graph theory algorithms, network reliability	THE
Dale Olesky	Numerical analysis, linear algebra	THE
Jianping Pan	Distributed systems, computer networks	SW
Frank Roberts	Numerical analysis, approximation theory	THE
Frank Ruskey	Combinatorial algorithms, discrete mathematics	THE
Micaela Serra	Hardware/software co-design, cellular automata	HW
Venkatesh Srinivasan	Computational complexity	THE
Ulrike Stege	Parameterized complexity, computational biology	THE
Margaret-Anne Storey	Human-Computer Interaction, e-Learning	SENG
Alex Thomo	Databases, data mining	SW
Melanie Tory	Human-Computer Interaction, Visualization	SENG
George Tzanetakis	Computer music, machine learning	SW
William Wadge	Dataflow, semantics, digital documents	SW
Jens Weber (Jahnke)	Middleware, interoperability	SENG
Kui Wu	Computer networks	HW
Jason Corless	Data compression, operating systems	SW
LillAnne Jackson	Memory consistency models	HW
Mary Sanseverino	e-Learning	
Michael Zastre	Programming languages	SW

The table above includes the four senior instructors (bottom 4 rows); their positions do not require them to conduct research and they receive no recognition in the salary review process for doing so unless the research enhances or is related to their teaching. However that does not mean that they have avoided conducting research.

4.1.1 FACULTY RESEARCH SUMMARIES

Mantis Cheng originally worked in logic and functional programming, but now focusses his attention on real-time and embedded systems, particularly as applied to mechatronics applications.

The main research interest of **Yvonne Coady** is aspect-oriented programming, particularly as applied to developing systems software. Her work involves programming language and software engineering techniques targeted at improving and evaluating large, complex systems.

Daniela Damian works in the areas of requirements engineering, computer-supported cooperative work and human-computer interaction. It involves collaborative aspects of software development, and in particular in studying processes and computer-supported collaborative tools.

The research of **Sudhakar Ganti** continues on from his industrial background in high-performance computer networks. He has particular interest in quality-of-service and performance issues.

Daniel German has varied research interests within the general area of software engineering. These interests include software evolution, open source software and intellectual property issues.

Daniel Hoffman has worked in automated testing for most of his career. He started with software testing in general, but has now focussed on testing computer networks, particularly their protocols.

Nigel Horspool has a long research record in systems software, especially compilers and the run-time implementation of object-oriented programming languages with virtual machines and just-in time compilation.

Jens Weber (Jahnke) has many interests loosely related to Software Engineering, including embedded systems and ubiquitous computing. He is also cross-appointed in the Department of Health Information Science and collaborates with them on medical informatics research.

Bruce Kapron conducts research in computational complexity, cryptography, and on the theoretical foundations of security.

Valerie King is recognized for her work on randomized algorithms; she also works on combinatorial algorithms and lower bounds, distributed algorithms, and applications to computational biology and networks.

Michael Miller has research interests in fault diagnosis in computer systems, VLSI design, and multiple-valued logic.

Hausi Müller's research interests include software engineering, software evolution, autonomic computing, adoption-centric software engineering, software architecture, reverse engineering, software re-engineering, program understanding, visualization, and software engineering tool evaluation.

Jon Muzio has a strong interest in logic design and switching theory, particularly as applied to fault tolerance and fault detection in computer systems.

Wendy Myrvold's expertise is in graph theory, graph algorithms, combinatorics and combinatorial algorithms. The topics she has investigated includes network reliability, topological graph theory, applications to chemistry, and Latin squares.

The work of **Dale Olesky** is in numerical analysis and related areas, including combinatorial matrix theory, linear algebra and graph theory.

Jianping Pan works in distributed systems and computer networks, and his recent research interests include protocols for advanced networking, performance analysis of networked systems, and applied network security."

Frank Ruskey specializes in combinatorial algorithms and discrete mathematics. In recent years he has been working extensively on Venn diagrams, Gray codes, and irreducible polynomials over finite fields.

Micaela Serra has worked in digital hardware testing; more recently the focus has been on hardware/software co-design and embedded systems.

Venkatesh Srinivasan holds a Tier 2 Canada Research Chair in complexity of computing.

Ulrike Stege researches in topics related to complexity theory, especially parameterized complexity theory and fixed-parameter-tractable algorithms. Much recent work has been related to algorithms for computational biology, as well as work in human problem solving.

Margaret-Anne (Peggy) Storey holds a Tier 2 Canada Research Chair in Human Computer Interaction (HCI). In addition to HCI, her research interests include program comprehension, software visualization and e-learning.

Databases and related problems, such as query optimization and data mining, form the main research interests of **Alex Thomo**. He also has interests in XML, internet programming, and string processing with applications to Biology.

Melanie Tory who only joined the department in April, 2006 has interests in computer supported collaborative work in addition to her primary interests in human-computer interaction and visualization.

George Tzanetakis specializes in issues related to the use of computers in music, including computer audition, audio signal processing, and music information retrieval.

William Wadge has wide ranging interests all loosely related by the study of semantics, including dataflow languages and other nonprocedural languages, as well as digital documents.

The main interest of **Kui Wu** is in computer networks. His research topics include mobile networks, sensor networks, wireless networks, network performance evaluation, and network security.

4.2 New Faculty Appointed in 2006

All three new faculty listed below work in the broad area of computer graphics.

Amy Gooch joined the department as an Assistant Professor on 1 September 2006. She was formerly a PhD student at Northwestern University, Illinois. She specializes in non-photorealistic rendering, as well as human perception in immersive environments and computer-generated and modified imagery. The research involves the interdisciplinary synthesis of computer graphics, perceptual psychology, art theory, and computational vision.

Bruce Gooch also joined the department as an Assistant Professor on 1 September 2006. He was formerly an Assistant Professor at Northwestern University, Illinois. His work combines computer graphics techniques for creating artistic imagery with the evaluation methods of perceptual psychology in order to enhance the communication capability of computer generated imagery.

Brian Wyvill was given a temporary part-time appointment with the department effective from 1 July 2006. His position became full-time as a Professor and holder of a Tier 1 Canada Research Chair on 1 January 2007. His research interests include computer animation, solid modelling, implicit modelling, and visualization.

4.3 Adjunct Faculty, Cross-Listed Faculty and Professors Emeritus

For the 2006/2007 academic year, the department has eleven adjunct faculty, four cross-listed faculty and five professors emeritus. The majority of these people are engaged in research activities, much of it in collaboration with faculty members in the department. They are listed in Tables 4.2, 4.3 and 4.4.

TABLE 4.2: Adjunct Faculty for 2005/2006

	External Organization / Employer
Ian Barrodale	Barrodale Computer Systems, Victoria
Ernest Chang	
Maurice Danard	
Brian Gaines	
David Goodenough	Pacific Forestry Centre, NRC
James McBride	
Morgan Price	Faculty of Medicine, UBC
Jackie Rice	Dept. of Computer Science, University of Lethbridge, Alberta
Dominique Roelants van Baronaigien	Dept. of Computer Science, Malaspina University College
Janice Singer	NRC Institute for Information Technology, Ottawa
Peter Walsh	Dept. of Computer Science, Malaspina University College

TABLE 4.3: Cross-Listed Faculty for 2005/2006

	Home Department at UVic
Peter Driessen	Dept. of Electrical & Computer Engineering
Benjamin Jung	School of Health Information Science
Andrew Schloss	Department of Music
Pauline van Den Driessche	Department of Mathematics

TABLE 4.4: Professors Emeriti for 2005/2006

	Research Interests
Byron Ehle	Numerical Analysis
John Ellis	Computational complexity, algorithms
Eric Manning	Distributed systems, quality of service, multimedia systems
Gholamali (Ali) Shoja	Distributed systems, multimedia systems
Maarten van Emden	Logic programming, constraint programming

4.4 Research Grant Funding

The one consistent source of externally funded research grants has been the NSERC Discovery Grants program. In July 2006, all but two of the 31 research faculty in the department held these grants. In addition, three researchers associated with the department as professors emeritus and four adjunct faculty hold Discovery grants. Table 4.5 shows the statistics for the research faculty in the department from 1999 to 2006.

The average award level has remained fairly constant over the seven years covered in the table. This reflects continued tightness in the NSERC budget available for Discovery Grants, and the addition of junior faculty to the department who would be expected to receive grant levels closer to the bottom end of the range as their first awards.

TABLE 4.5: NSERC Discovery Grants, 1999-2006

	1999	2000	2001	2002	2003	2004	2005	2006
Total	$447.7K	$494.2K	$511.2K	$568.3K	$605K	$666.9K	$728K	$736.9K
Awards	19	20	20	23	24	27	29	29
Average	$23.6K	$24.7K	$25.6K	$24.7K	$25.2K	$24.7K	$25.1K	$25.1K

Research grants from other sources tend to have a short duration, often a one-time award in a single fiscal year, and show wide variations in value. It would be unwise to extract any long-term trends from the data for such awards. The numbers are reported in Table 4.6.

TABLE 4.6: Research Grants and Contracts, excluding Discovery Grants, 1999-2005

	1999	2000	2001	2002	2003	2004	2005	2006
Total	$712K	$730K	$978K	$1112K	$1159K	$479K	$680K	$1259K
Awards	12	12	17	18	22	10	15	14
Average	$59.4K	$60.8K	$57.5K	$61.8K	$52.7K	$47.9K	$45.3K	$89.9K

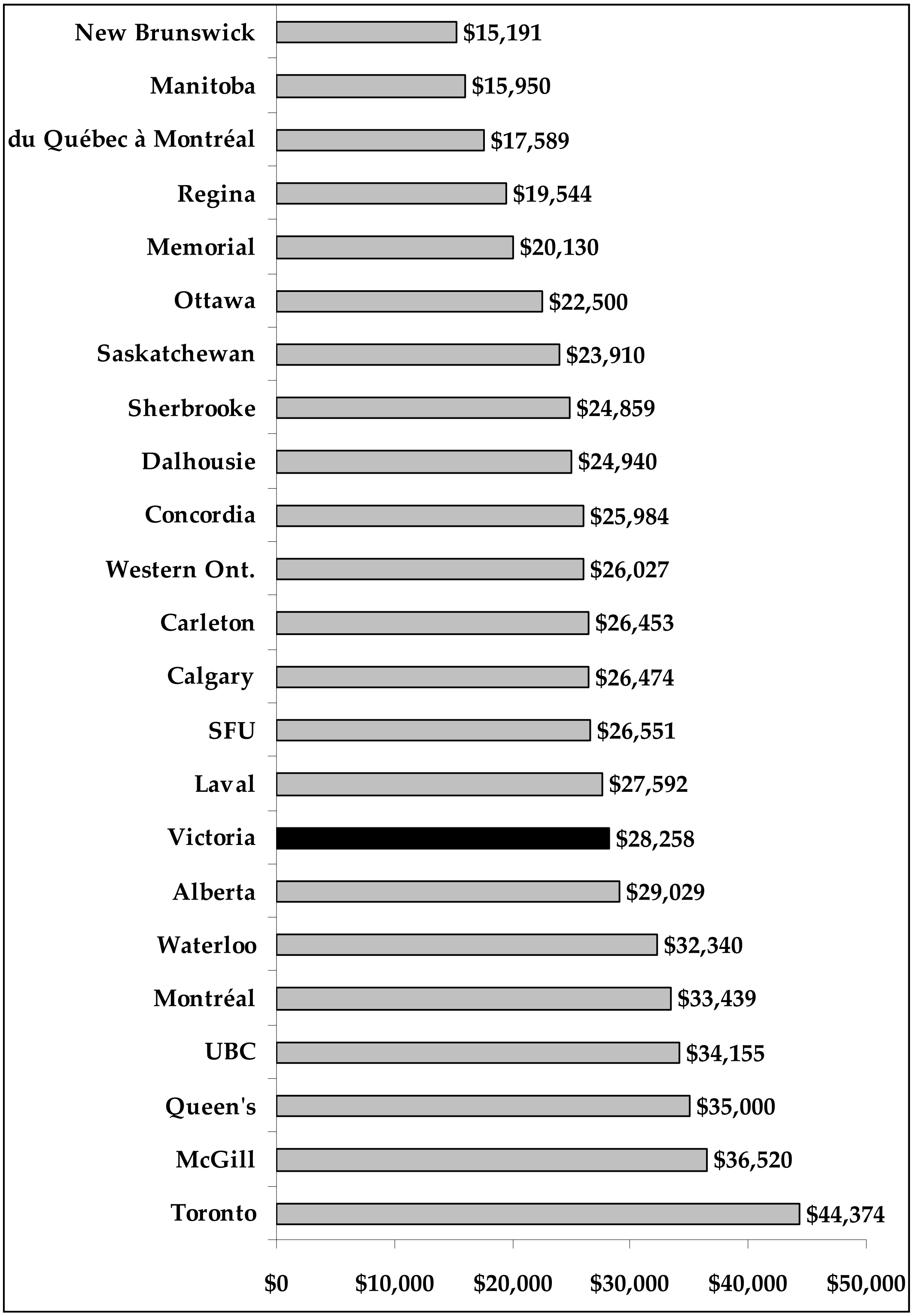
Grants for equipment purchases vary even more erratically from year to year, with large CFI grants dominating the numbers in recent years. The totals for equipment grants are shown in Table 4.7.

TABLE 4.7: Equipment Grants, 1999-2005

	1999	2000	2001	2002	2003	2004	2005	2006
Total	$191K	$177K	$5K	$97K	$679K	$182K	$484K	$495K
Awards	6	2	1	4	5	3	6	6
Average	$31.9K	$88.6K	$5.0K	$24.2K	$135.9K	$60.6K	$80.7K	$82.5K

The average level of NSERC Discovery Grant awarded to computer science researchers at UVic, as compared to other computer science departments across Canada, provides some validation that the department is doing well. According to the annual Taulbee Report published by CRA, there are 23 PhD-granting computer science departments in Canada. We have used NSERC data to calculate the average Discovery Grant award for these 23 departments and tabulated the results in Figure 4.2. In 2005, our department ranked 8th amongst these 23 departments.

FIGURE 4.2. Average 2005/2006 Discovery Grant compared by University



4.5 Research Collaborations

4.5.1 RESEARCH COLLABORATIONS WITHIN UVIC

Faculty members participate in many collaborative research projects with others both within and outside the university. Internally the primary collaborations are:

- *Music:* George Tzanetakis
- *Psychology:* Daniela Damian, Ulrike Stege, Peggy Storey
- *Mathematics:* Dale Olesky
- *Biology:* Ulrike Stege, Valerie King
- *English:* Daniel German
- *Health Information Science:* Jens Weber (Jahnke)

Over the past few years researchers in the department have collaborated with a large number of people in other institutions and organizations. A sample of some of these follows:

- Australian National University, Computer Science
- Bell Labs, Lucent Technologies, Naperville, IL, U.S.A.
- British Columbia Institute of Technology
- Carnegie Mellon Software Engineering Institute, Pittsburgh
- Case Western Reserve University, Computer Science
- Cogneto Development, Vancouver, British Columbia
- Concordia University, Montreal
- Curtin University, Australia, Computer Science
- Dalhousie University
- Dell, US and Brazil
- Enquisite, Victoria, British Columbia
- HP Labs, Bristol, England
- Humboldt University Berlin, Germany, Institüt für Wirtschaftsinformatik
- IBM Center for Advanced Studies, Toronto, Ontario
- IBM Ottawa Software Lab
- IBM, T.J. Watson Research Center
- Indian Institute of Science, Bangalore, India.
- InfoClin, Toronto, Ontario
- Intec Automation Inc, Victoria, British Columbia
- Inter Tribal Health Authority, Nanaimo, British Columbia
- McMaster University Computing and Software
- Macalester College, MN., Mathematics
- Memotrax Inc, Vancouver, British Columbia
- National Research Council, Ottawa
- New York University
- North Carolina State University, Computer Science
- Northern Health Authority, Prince George, British Columbia
- NRC, Canada

- Orcalab, Hanson Island, British Columbia (http://www.orcalab.org)
- PG Music Inc, Victoria, British Columbia
- Pontificia Universidade Catolica do Rio Grande du Sul (PUCRS), Brazil
- Princeton University
- Purdue University
- San Jose State University, Mathematics
- Siemens Research, USA and PSE Austria
- Simon Fraser University, Computing Science
- Stanford University, Computer Science
- Suffolk University, Boston
- Teligence Inc, Vancouver, British Columbia
- Sun Microsystems of Canada, Vancouver, British Columbia
- Technical University of Eindhoven, The Netherlands
- Technical University of Munchen, Germany
- TU Berlin, Germany, Computer Science
- Unisys Australia
- Universitat de Lleida, Spain
- University of Alberta
- University of Ballarat, Australia, Computer Science
- University of Bari, Italy
- University of British Columbia, Computer Science
- University of British Columbia, Civil Engineering
- University of British Columbia, Family Practice
- University of Calgary, Computer Science
- University of Florida, Gainseville, Mathematics
- University of Kassel, Germany, Computer Science
- University of Manitoba, Computer Science
- University of New Brunswick, Fredericton
- University of Newcastle, Australia, Computer Science
- University of Sheffield, England, Chemistry
- University of Sydney, Australia
- University of Toronto, Computer Science
- University of Toronto, Mechanical and Industrial Engineering
- University of Waterloo, Computer Science
- Victoria Hospice Society, Victoria, British Columbia
- Vienna University of Technology, Austria
- Vigil Health Solutions Inc., Victoria, British Columbia

4.6 Undergraduate Involvement in Research

Undergraduate students play an important role in research conducted within the department. Their participation typically takes one of these forms:

- employment as a research assistant

- taking a directed studies course (CSC 498) with a faculty member where the course has a project which develops into research
- completing a Honours project (CSC 499) for the BSc (Hons) program, where the project assigned by a faculty member is research quality
- receiving a NSERC USRA award and working with a NSERC funded researcher on a research project.

Over the last seven years, the number of NSERC USRA awards taken within the department has consistently been 6 or 7 each year.

Many undergraduates have authored or co-authored papers as a result of their involvement in research projects. These students include:

- Richard McWalter, Manj. Benning, and Stuart Bray (with G. Tzanetakis)
- Alison Meynert, Daniel Horspool, S. Girn, and M. Minchenko (with W. Myrvold)
- Brad Bingham, Aidan Roy, and T.X.T. (Dick) Phan (with D. Olesky)
- Jeff Proctor, Steve Langerhan, Owen Stampflee, Marco Yuen, Stuart Bray, and Celina Gibbs (with Y. Coady).

4.7 Publications and Other Measures of Research Impact

While counting publications provides an imperfect measure of research productivity, there are no better alternatives which are easily available. Table 4.8 shows the combined count of publications in three forms: contributions to books, refereed journal publications and refereed conference publications, as reported in the faculty curriculum vitae for each year from 1999 to 2005. The average over the entire 7 year period is 4.2 refereed publications per faculty member per year.

TABLE 4.8: Publication Counts for Faculty

Year	1999	2000	2001	2002	2003	2004	2005
Publications	67	69	94	103	115	95	106
No. of faculty	18	19	22	23	23	26	28
Pubs/faculty	3.72	3.63	4.94	4.68	5.00	3.65	3.79

More important than the quantity of research publications is the *impact* of that research. Perhaps the number of citations for the publications could be used as a measure of impact. However, it is remarkably difficult to collect this information and even more difficult to obtain comparative information for other computer science departments. Instead, we choose to provide some anecdotal evidence of the impact of research conducted in the department. Some highlights are:

- George Tzanetakis published the paper “Musical Genre Classification for Audio Signals” in IEEE Transactions on Speech and Audio Processing, Volume 10, Number 5, July 2002, for which he won the prestigious 2004 Young Author Award from the IEEE Signal Processing Society. It is widely cited and considered one of the seminal papers in the area of Music Information Retrieval.

- Daniela Damian won the best paper award at the IEEE International Conference on Requirements Engineering 2000, for her paper “Using Different Communication Media in Requirements Negotiation”.
- The work of Frank Ruskey and his students is featured in several places in the recently published fascicles of volume 4 of Donald Knuth's famous series of books, “The Art of Computer Programming”.

Many members of the department have been very heavily involved in professional activities and contributions to the overall strength of the discipline. In the last three years, faculty members have served on 99 conference program committees and 17 conference and workshop organizing committees. Two people have served as conference program chairs and seven have acted as general chairs of conferences and workshops. In addition, ten faculty have acted as editors of special issues of journals, as well as five faculty members serving on editorial boards and as associate editors of journals.

5: *Ancillary Activities*

5.1 Equity

The primary efforts of the department have been concerned with increasing the recruitment and retention of women in our programs. These initiatives are described below. The department has also attempted some initiatives under the leadership of Dr. Kapron looking at the issue of aboriginal students in our programs, but we have encountered the same difficulties as other science and engineering departments in this regard and many such students are, at present, ill prepared to take our programs.

In this section all the figures with respect to Canada come from the 2007 CAUT Almanac.

5.1.1 UNDERGRADUATE STUDENTS

The percentage of undergraduate female students is as follows:

	Canada: 2004-05	UVic, 2004-07
CSC & CIS	15.6%	15.4%

The above percentages for UVic do not include any of the students who are in combined programs as these numbers are very difficult to get as all of these programs formally reside in other faculties – there are about 50 such students in 2006-07.

5.1.2 GRADUATE STUDENTS

The percentage of undergraduate female students is as follows:

	Canada: 2004-05	UVic, 2004-07
CSC & CIS	23.8%	26.1%

5.1.3 FACULTY

The department is very proud of its record in hiring outstanding female faculty members over the last twenty years.

	Canada 2004-05			UVic 2006-07		
	Total number	Female number	Percentage	Total number	Female number	Percentage
Full	324	27	8.3%	12	3	25%
Associate	315	60	19.5%	4	1	25%
Assistant	369	69	18.7%	14	5	35.7%
Other	93	27	29%	4	2	50%
Total	1101	183	16.6%	34	11	32.4%

5.1.4 STAFF

	Male	Female	Total
Office support staff	0	5	5
Administrative support staff	0	3	3
Teaching support staff	1	2	3
Technical support staff	5	2	7
Total	6	12	18

5.2 Outreach and Recruitment

5.2.1 WOMEN IN ENGINEERING AND COMPUTER SCIENCE (WECS)

Following a realization in 2002 that the percentage of women in computer science programs at UVic was lower than that at most other Canadian universities, a proposal was made to the senior administration for a pilot program to encourage the recruitment and retention of women in our computer science programs. The pilot project started in 2003 and included a half time staff person paid from non-recurring funds (*soft money*).

This program has now expanded to include a full time coordinator in base budget and its role has expanded to encompass the whole of the Faculty of Engineering.

The primary current activities can be characterized as of the following three types:

- *High school visits.* There are a number of different presentations given in the schools.
- *Workshops.* One day workshops which are given both at UVic (for girls from the local community) and at various schools around the province. Recently they have been given in Cranbrook, Fernie, Gabriola Island, Powell River, Van Anda, Saltspring Island and Texada Island.
- *Community building activities for current female students.* These include pizza parties, coding competitions, barbecues and mentoring. There is an attempt to involve as many of our alumni as possible in such events.

Over the last few years, many different initiatives have been tried and the most successful have been the *Lego Mindstorms Workshops*. These have been very well received and are particularly popular – we have had trouble being able to run a sufficient number of them. We started out running one such workshop per term, but the success has been such that we are running three in April 2007, for example.

In these workshops, the girls learn to be confident with technology such as robots, learn about the parallels between the concepts they explored and the technologies they use regularly such as text messaging, mp3 players, games, etc. They also learn that math skills are crucial to success in Engineering and Computer Science. The publicity around our Lego Mindstorms Robotics Festival was picked up across North America, being circulated by a faculty member at MIT to all board members of CRA. It was also featured in the ACM Tech News.

5.2.2 OTHER OUTREACH AND RETENTION

The department is in the middle of a transition from more passive outreach to very active outreach. Outreach used to be primarily volunteer activities for department members, but with the changing times, it has become critical for all members. This does put an increased expectation on faculty members. While the University is very supportive of all recruitment activities, it has not, so far found any direct budget support for such activities in the department (apart from WECS). The Faculty of Engineering has a recruitment officer, but there are considerable differences between the professional mandatory co-op engineering programs and the various computer science programs.

So far, we have been unable to publicize our new combined programs to the level that is needed to increase awareness of them. This makes it very difficult to meet the enrolment targets that the University has set for the department.

All of the usual outreach activities are undertaken by members of the department including informal talks given by members in the local community, and extensive participation in the Speakers Bureau run by the University. We do, of course, have extensive contacts in the local industry and government though our co-op program. We also have the clear advantage over many universities that the co-op program is integrated very closely with the department, with our coordinators being full members of the department.

The department was heavily involved in the very successful open house held in connection with the official opening of the ECS building. In addition, the department is an active participant in Science Venture and Go WEST. Science Venture offers eight week summer camp programs as well as workshops and community events during the year. The goal is to offer hands-on, minds-on science, engineering and technology learning opportunities for kids entering grades 1 through 12. Go WEST offers an assortment of summer programs including residential conferences and is designed to promote engineering to secondary students.

For the last six years the support staff, under the leadership of Bill Gorman and Brian Douglas, have been doing lab tours, in the form of workshops during inter-session, reading breaks and open houses. These have been targeted primarily but not exclusively, at high school and middle school students.

These activities have included Experiments in Collaborative Computing, Robot Demo, Selected Java Programming Tasks and Demonstrations, Museum of Technology, Building an Animation and most recently Alice: 3D Visual Programming environment. Since the beginning we have steadily increased the number and variety of demonstrations as part of the staff effort towards Outreach and increasing our enrollments.

In terms of retention, the department is very successful once students are into second year. However, we have been losing a much large number from first year than we would like, so we are attempting to improve the retention of students from first year in our programs in a number of different ways. The most important was for us to redesign our first year courses (CSC 110 and 115) to be much more effective and interesting for the students. It had become clear that, for many students, CSC 115 was a very intimidating course and that we were attempting to include too much material. This revision was undertaken by a task force that considered all aspects of the first year program and considerable changes have been made to these courses and their associated labs in the last year.

5.3 Professional Activities

5.3.1 INVITED TALKS AND PRESENTATIONS

Members of the department have been very active in giving keynote talks at conferences and invited talks at other institutions, for industry and government. They have given over 200 such talks in the last three years.

5.3.2 EXTERNAL PROFESSIONAL CONTRIBUTIONS

Members have taken a very active professional role, with faculty serving on NSERC grant selection committees and other NSERC panels, as well as acting as editors of special issues of professional journals, serving on editorial boards of journals, holding office in professional societies, and acting as chairs of workshops and conferences.

A complete list of all such activities would be too long to be included here. The information can be gathered by reading the faculty CVs. It must be emphasized as a summary that, over the last three years, faculty members have served on more than 35 organizing committees for conferences, as well as more than 140 program committees for conferences and workshops.

5.3.3 ADMINISTRATIVE STRUCTURE

The committee structure of the department and the terms of reference of each committee are listed below.

ADVISING

Advisor	Jane Guy
Co-op advising:	T. Garrett, M. Casey
Honours advising:	D. Olesky
Transfer credits:	F. Roberts (31 Dec. 2006)/L. Jackson (as of 1 Jan. 2007)
Record of Degree Program (RDPs):	Jane Guy

COMMITTEES

Academic Integrity Committee

M. Zastre (Chair), A. Gooch, J. Guy, V. Srinivasan, Chris Luft (CSCU Rep.), Warren Shenkenfelder (Grad Student Rep.)

Mandate:

- recommend department policies in all areas concerned with academic integrity;
- investigate and recommend actions to be taken in connection with violations of academic integrity within the department.

CSc/Math Co-op (a joint committee of Computer Science and Math):

D. Olesky, (Chair), M. Bultmann, M. Casey, T. Garrett (Co-Op); J. Huang, B. Khouider (Math); W. Myrvold (CSc).

CSc Admissions Representative: Dale Olesky

CSc Retention and Placement Representative: W. Myrvold

Mandate:

- represent the department in the coordination and operation of the CSc/Math Co-op program.

Combined Programs:

U. Stege, (Chair), J. Guy, B. Gooch, A. Thomo, Dale Cope (CSCU Rep.)

Mandate:

- oversee and advise on revision of combined programs;
- coordinate outreach and publicity for combined programs.

Equity Committee:

M. Tory (Chair), B. Gooch, W. Myrvold, Jonathan Cobb (CSCU Rep.)

Mandate:

- oversight of departmental equity issues in regard to: recruitment, training and development, promotion, retention, accommodation, consultation and monitoring;
- the committee should liaise with the UVic Office of Equity Issues.

Graduate Advisor:

M. Serra

Mandate:

- Day-to-day administration of graduate program, including:
    - changes to student record including degree program, supervisory committee and registration;
    - requests for oral examination
    - monitoring performance of graduate students;
    - ongoing support arrangements for graduate students;
    - orientation for new students;
    - providing consultation to students.

Graduate Studies Committee:

M. Serra (Chair), J. Pan, F. Ruskey, A. Thomo, W. Wadge, P. Rigby (Grad. Rep)

Mandate:

- recommendations for awards administered by Faculty of Graduate Studies;
- changes to student record including degree program, supervisory committee and registration;
- monitoring performance of graduate students;
- recruitment of graduate students;
- graduate curriculum.

Graduate Admissions: M. Serra

Mandate:

- overseeing the process of graduate applications;
- initial support arrangements for graduate students.

Graduate PhD Breadth Committee:

Same membership and Chair as for Graduate Studies Committee.

Mandate:

- administer the PhD breadth requirement;
- when appropriate, the Committee may request help from other faculty; members in evaluating a student's breadth. It is also responsible for revisions to the breadth requirement rules when needed.

Outreach, Communication and Publicity:

M. Sanseverino (Chair), M. Cheng, S. Reiser, A. St. Pierre, U. Stege, B. Wyvill, David Sprague (Grad Student Rep)

Mandate:

- Recommends in all areas concerned with external activities of the department, apart from those covered by the World Wide Web/Internet Infrastructure Committee. These include:
  - external publicity and advertising;
  - liaison with the Women in Computer Science program;
  - student recruitment;
  - other outreach initiatives.

Research Infrastructure (RIC):

N. Horspool (Chair), S. Ganti, R. Taylor, A. Trumpour, Dale Lyons (Grad Student Rep)

Mandate:

- Planning and administration of Departmental facilities in support of research including:
  - long term planning;
  - seeking additional resources through grant applications;

Teaching Infrastructure (TIC):

M. Serra (Chair), L. Jackson, D. Church, N. Horspool, A. Trumpour, Jim Roepcke (CSCU Rep)

Mandate:

- reviewing current policy and procedures for allocation of resources (physical, human and financial) in support of teaching at both the undergraduate and graduate levels;
- planning and administration of Departmental computing facilities in support of teaching;
- maintaining a prioritized list of equipment and software requirements (which we would be authorized to purchase after reasonable consultation with no formal debate and vote at a full Department Meeting).

Undergraduate Curriculum:

J. Guy/M. Zastre (Co-Chairs), J. Corless, T. Garrett, W. Wadge, K. Wu, Jim Roepcke (CSCU Rep)

Mandate:

- initiate and review undergraduate course and program changes;
- liaison with other units regarding undergraduate curriculum including joint/combined programs;
- responsibility for undergraduate calendar entries;
- attendance at ECE Undergraduate Curriculum Committee meetings.

Women in Engineering and Computer Science Liaison Committee (WECS)

M. Sanseverino (Chair), D. Damian, U. Stege, S. Reiser, A. St. Pierre, Eleanor Wang (CSCU Rep)

World Wide Web/Internet Infrastructure:

M. Zastre (Chair), D. German, J. Weber, P. Stead

Mandate:

- design of the Departmental web pages (and other Internet resources) that are accessible to the outside world;
- overseeing implementation of Departmental web pages;
- ensuring that the content of Departmental web pages remains up to date.

CSC 110/115 Redesign Team:

I. Bull, Y. Coady, J. Corless, B. Gorman, L. Jackson, W. Myrvold, A. Thomo, M. Zastre

Task Force to redesign CSC 212:

M. Sanseverino (co-Chair), U. Stege (co-Chair), B. Gooch, J. Ryall, A. Thomo

Task Force to redesign CSc/Psychology Combined Programs:

J. Guy, J. Muzio. U. Stege

Task Force to review CSc/Visual Arts Combined Program:

U. Stege, A. Thomo

5.3.4 DEPARTMENT REPRESENTATIVES

Computer Science Course Union Liaison:	M. Zastre
ECE Undergraduate Curriculum:	(S. Ganti), D. Hoffman (from 1 Jan. 2007)
Library:	K. Wu
Mechatronics:	M. Cheng
Old Equipment Curator:	B. Douglas
Provincial Articulation Committee:	J. Corless, M. Zastre
Science and Engineering Subcommittee for UVic Research and Travel Grants:	W. Myrvold
Special Students Program:	T. Garrett

5.3.5 FACULTY COMMITTEES

BENG/BSENG Co-Op Committee:	J. Muzio, D. Olesky (as designate)
BSENG Curriculum Committee:	H. Müller (Chair), M. Cheng, M. Zastre
Ceremonies Committee:	D. Olesky
Curriculum Committee:	M. Zastre (30/6/2007)
Engineering & Science Speaker's Bureau:	M. Serra
Executive Committee Member:	J. Muzio
Faculty Advisory Committee to the Dean:	N. Horspool (1 Jul. 06 - 30 June 08)
	D. Olesky (1 Jul. 06 - 30 June 07)
	M. Serra (alternate) (1 Jul. 05 - 30 Jun. 07)
Faculty Evaluation Task Force (FETF):	J. Corless, F. Ruskey, J. Weber
MECH Promotion Committee:	M. Serra
Search Committee for CSc Chair:	B. Gorman, D. Hoffman, W. Myrvold, M. Serra, V. Srinivasan
Search Committee for BSENG Program Director:	D. Hoffman, M.-A. Storey
Search Committee for Associate Dean:	H. Müller, K. Wu
SENG Program Board:	H. Müller, J. Muzio, J. Weber
Student Recruitment Committee:	M. Sanseverino
Strategic Planning Committee:	H. Müller, J. Weber
Undergraduate Programs Coordination Committee:	T. Garrett, J. Muzio, H. Müller

5.3.6 UNIVERSITY COMMITTEES

The department is very well represented by service on university wide committees.

Advisory Committee to the Learning and Teaching Centre:	J. Muzio, M. Sanseverino, M. Storey
Bicycle Users Committee:	M. Sanseverino
Diversity and Equity Committee:	M. Tory, A. St. Pierre
Faculty/Co-op Council:	D. Olesky
Faculty Women's Caucus Steering Committee:	D. Damian, M. Sanseverino
Instructional Technology Working Group, Learning and Teaching Centre:	D. German, M. Sanseverino
NOVA Project Committee:	J. Muzio
PIMS Steering Committee:	F. Ruskey
Personal Computer Acquisition Enhancement Fund:	J. Muzio
President's Committee on Distinguished Lectures:	M. Serra
Senate Committee on Awards:	D. Olesky
Senate Committee on Appeals:	M. Serra
Senate Committee on Committees:	M. Serra
Senate Planning Committee:	D. Hoffman
Senior Instructor Advisory Committee to Faculty Assoc.:	M. Sanseverino
University Club Director:	N. Horspool
University Women's Lecture Series Committee:	V. King

6: Administrative and Teaching Support

There are 18 support staff employed by the department, divided into four groups.

6.1 Departmental Office Support Staff

Department Secretary:	Sharon Moulson
Graduate Secretary:	Wendy Beggs
Secretary:	Nancy Chan
Secretary:	Isabel Campos
Receptionist/Secretary:	Carol Harkness

There are four full-time positions in the main office and one part-time (.79 FTE) position. During the past six years the receptionist position was upgraded to full-time position and filled by Carol Harkness. Nancy Chan's position was upgraded from 0.50 FTE to 0.79 FTE and the duties were rewritten to include research clerical support for the department's research faculty and their graduate students. This position is currently under review for a possible re-classification. The Graduate Secretary's position was reclassified to a PB7 from a PB6. It should be noted that with the exception of the Graduate Secretary position, the turnover in the office staff has been non-existent with the majority of staff being in their positions since 1994. While the Graduate Secretary position is viewed as a training ground for higher level positions, it has become apparent that the workload in this position is significantly higher than in other comparable positions across campus, as computer science has the largest thesis-based graduate program on campus.

Over the next five years, it is not anticipated that additional support staff in the main office would be needed especially if Nancy Chan's position was increased to full-time. With this increase, the Departmental Secretary could assign to this position some of the duties associated with maintaining the departmental accounts. It is also recognized that most research faculty handle most of their research administrative tasks themselves.

If the two administrative officers, the Women in Computer Science (WECS) program or research faculty require more administrative support than they are currently asking for, it is possible another part-time support position would be required. Also, it would depend on how many more graduate students the department would admit in the next five years. If it becomes substantially more than the 131 students we have at present, then additional support for the graduate program would be necessary.

6.2 Administrative Support Staff

Undergraduate Advising Officer:	Jane Guy
Administrative Officer:	Susanne Reiser
Coordinator, Women in Engineering and Computer Science:	Anissa St. Pierre

Prior to 2003, the department had only an Administrative Officer, who also undertook some advising, but the increasing workload required the addition of a second person. The Undergraduate Advising Officer advises CSc students on their programs, course selections, and a wide variety of academic problems, assisting in course registration, prerequisite waivers, and managing transfer credit evaluation. She also processes degree declarations and does graduation checks prior to convocation. In addition, she develops academic timetables, monitors and forecasts enrollment, and prepares demographic reports. She also administers student acceptance to Combined Music/CSc program and is an active member of many department committees (Curriculum, Academic Integrity and the Combined Programs Committees).

The Administrative Officer, in addition to all of the usual administrative duties, also serves as a back up undergraduate advisor as required and has an increasing role as a communications officer for the department.

The work of both of these positions was seriously disrupted in the three years prior to June 2006 by the planning, building and moving into the new ECS building.

6.3 Undergraduate Teaching Support Staff

Senior Lab Instructor:	Bette Bultena[a]
Senior Lab Instructor:	Victoria Li
Lab Instructor and Senior Consultant:	Bill Gorman

[a] B. Bultena is on leave, 2006-07; the position is temporarily filled by D. Church

In addition to the permanent staff, the department also fills approximately 80 positions each term as lab instructors, markers, and help desk consultants. These positions are normally filled by graduate students plus, when required, a small number of upper level undergraduate students.

The three Senior Lab Instructors work as a team, with the following responsibilities:

- lab management, including software and hardware support that is not covered by the technical support team
- active support in course development for the core courses
- maintenance of supplemental course materials

- maintenance of training materials, forms, and schedules
- training and supervision of all of the graduate students employed each term as lab instructors, markers and consultants at the help desk
- supplemental tutorials for the students, with one-on-one assistance for student assignments
- instruction, marking, and leading lab courses
- development and delivery of outreach seminars
- assist in the redevelopment of courses, labs and curriculum.

With the redevelopment of our curriculum, courses and labs, the pressure on the three permanent staff has increased considerably.

6.4 Technical Support Staff

Senior Programmer Analyst:	Allan Trumpour
Senior Programmer Analyst:	Paul Stead
Senior Systems Administrator:	Robert Taylor
Programmer/Analyst:	Jill Aschenbrenner
Programmer/Analyst:	Kathryn Wilson
Programmer/Consultant:	Tomas Bednar
Electronics Technician:	Brian Douglas

Temporary technical support staff are employed as required (for example, such an occasion occurred with the move to the new building).

Before 1998, the CSc faculty offices were in the Engineering Office Wing (EOW) and the research labs and graduate students offices were in the Engineering Lab Wing (ELW). For the three years from 2003 - 06, the research labs and graduate students space were all located in TEF, a building about 10 minutes walk across campus. Since May 2006, all of our faculty offices, research labs, teaching labs and graduate student offices have been housed in the Engineering/Computer Science (ECS) building. For the past four years, there have been tremendous changes in the computing and networking infrastructures, creating additional workload for our technical support staff.

Currently, we have 10 teaching labs and 12 research labs. We support roughly 30 Unix servers, 8 Windows servers, 200 Unix workstations, and 500 Windows workstations, divided evenly between teaching and research. The 10 teaching labs include the computer graphics and multimedia lab, the mechatronics and embedded software lab, the Linux and networking lab, the architecture and digital logic lab, and several general-purpose Windows labs. Our research labs include the computer graphics lab, the human/computer interface lab, the distributed collaboration lab, the theory lab, the automated software testing lab, the remote sensing and imaging lab, the computer music lab, the network performance lab, and the reverse engineering lab. Although the networking (wired and wireless) infrastructure is under the responsibility of the

Faculty of Engineering support staff, the basic networking configurations and services are managed by CSc technical staff.

We have six full-time technical support staff for all software and networking issues, and one full-time technician taking care of all hardware and installation issues. They spend roughly 30% of their time on research, 50% on teaching and 20% on administrative infrastructure support. The services that our technical staff provide include mail and SPAM filtering, file storage and backup, printing, web, user accounting, computer security and firewalls, software application installation and updates, and hardware installation and repair. They also handle network configuration and DHCP management, secure access card control, and general IT support. Each term, after all course related projects and assignments have been completed, all shared workstations must be “re-imaged”, i.e., new disk images are installed to remove all stale files and configurations.

In the past six years, we have acquired five new Unix servers, and four mail/firewall servers for both teaching and research support. Most of our faculty members and graduate students have their own workstations and laptops; each may have hundreds of gigabytes of hardware storage. It has become quite challenging to provide a day-to-day backup service, both in terms of storage media as well as network bandwidth issues.

It is also noted that it is difficult to predict what impact Project NOVA and the many administrative tasks that will change with its implementation will have on support staff workloads.

7: Resources and Facilities

7.1 Personnel

7.1.1 FACULTY

The number of regular faculty in the department has increased rapidly in the last five years with a net increase of ten positions. Of these ten positions, three are Canada Research Chairs, two are senior instructors and the other five are research faculty positions. The primary reason for the increase has been the provincial government's “Doubling The Opportunity” initiative designed to increase the numbers of graduates in electrical and computer engineering and in computer science.

Years	Regular Faculty	Years	Regular Faculty
1993/94	16	2000/01	24
1994/95	18	2001/02	24
1995/96	19	2002/03	27
1996/97	18	2003/04	27
1997/98	21	2004/05	31
1998/99	22	2005/06	31
1999/00	23	2006/07	34

Regular research faculty normally teach three courses per year if they have an active research program, otherwise they teach one or two more courses depending on the level of their administrative activities. In addition, it is assumed all research faculty will be supervising graduate students, and take a normal part in the administrative activities of the department.

There are three Canada Research Chairs in the department and they receive a one course teaching reduction. Similarly the two faculty members who have NSERC University Faculty Awards also receive a one course teaching reduction during the tenure of their award.

It is possible for faculty to buy out of teaching one course when there are reasons for so doing, but no faculty member is allowed to reduce their teaching level to zero.

Senior Instructors who do not have responsibility for conducting a research program normally teach six courses per year plus other associated duties in connection with the labs and similar.

There are also possibilities for course reductions for administrative duties - at present these are given to the Chair, the Director of Undergraduate Studies, and the Graduate Advisor.

The distribution of positions is as follows:

Position	Number of faculty
Research Faculty - Assistant	14
Research Faculty – Associate	4
Research Faculty – Full	12
Teaching Faculty	4
Emeritus Faculty	5
Adjunct Faculty	11
Cross-Listed Faculty	4

Regular faculty demographics

At the present time, there is some uncertainty with regard to mandatory retirement which is still in place at the University of Victoria but may possibly be abolished in the near future. The demographics of the regular faculty are distributed as in:

Age 40 or less:	11
Age from 40 - 50:	8
Age from 50 - 60:	11
Age 60 or over:	4

7.1.2 SESSIONAL INSTRUCTORS

In this category, we include a number of types of non–regular faculty who are teaching courses in the department. This includes people with short term visiting appointments, faculty who are members of the department while on sabbatical from their home institutions, adjunct faculty, individuals who have permanent positions elsewhere (usually in local industry) but have the requisite skill set as well as some of our graduate students.

The percentage of courses which cannot be covered by regular faculty has decreased over the last few years as new faculty have been hired, but we would anticipate an ongoing need for non regular faculty teaching at around 20% of the course offerings. Table 7.1 shows the statistics for course sections as taught by regular or sessional faculty.

TABLE 7.1: Course sections taught per year (May 1 – April 30)

	Regular Faculty	Non-regular faculty	Total	% Non-regular
2001/02	54	35	89	39%
2002/03	67.5	29.5	97	30%
2003/04	64	32	96	33%
2004/05	89	28	117	24%
2005/06	81	30	111	27%
2006/07	74	21	95	22%

7.2 Budget

7.2.1 OPERATING FUNDS

The major divisions of expenses in the budget are indicated in Table 7.2. As it is the case with most academic units, salaries are the major portion of the overall budget. The information is given only for the past 6 years and thus does not explicitly include the fluctuations prior to the peak enrolment years.

TABLE 7.2: Operating funds and their categories

	2000/01	2001/02	2002/03	2003/04	2004/05	2005/06
Faculty Admin/ Prof Salaries	$2,335,273	$2,349,916	$2,534,351	$3,155,752	$3,498,519	$3,393,685
Specialist/ Instructional Salaries	$426,655	$438,565	$456,479	$480,281	$472,677	$455,588
Total Teaching Salaries	$2,761,928	$2,788,481	$2,990,830	$3,636,033	$3,971,196	$3,849,273
Exempt/Office/ Tech. Salaries	$199,576	$225,635	$340,556	$261,279	$264,075	$261,330
Total Personnel Salaries	$2,961,504	$3,014,116	$3,331,386	$3,897,312	$4,235,271	$4,110,603
Equipment and Furnishings	$105,918	$154,816	$168,960	$160,203	$143,431	$108,001
Supplies and Expenses	$122,648	$142,547	$165,446	$230,802	$252,132	$204,349
Other	$405,044	$412,814	$453,472	$571,463	$582,223	$602,150
Total Expenditures	$3,595,114	$3,724,293	$4,119,264	$4,859,780	$5,213,057	$5,025,103
Annual Percentage Increase	8.76%	3.59%	14.58%	17.98%	7.27%	-3.61%

7.2.2 COMPARISON WITH OTHER UNITS

While every unit in a university has its own characteristics and thus the budgets are not easily comparable, it remains instructive to glance at the distribution of funding among the departments in Science and Engineering, as shown in Table 7.3.

TABLE 7.3: Comparison of operating budgets with other UVic units

	2000/01	2001/02	2002/03	2003/04	2004/05	2005/06
CSC	$3,595,114	$3,724,293	$4,119,264	$4,859,780	$5,213,057	$5,025,103
ECE	$2,765,637	$2,728,340	$3,212,074	$3,580,348	$4,023,396	$4,100,202
MECH	$2,435,760	$2,259,309	$2,282,749	$2,743,996	$2,990,806	$2,917,509
BIOC	$2,015,735	$2,051,303	$2,093,282	$2,720,662	$2,496,271	$2,699,374
BIOL	$3,603,619	$3,510,162	$3,684,466	$3,648,735	$4,075,306	$3,745,404
CHEM	$3,290,905	$3,373,997	$3,937,161	$4,016,399	$3,823,278	$4,209,186
EOS	$1,598,201	$1,635,909	$1,664,685	$1,965,760	$1,954,993	$2,082,134
MATH	$3,067,840	$3,265,506	$3,475,390	$3,672,009	$3,644,935	$3,816,147
PHYS	$2,916,592	$3,055,745	$3,365,238	$3,784,157	$3,493,273	$3,670,727

7.3 Space

7.3.1 STUDENT SPACE

The department is delighted to have moved into its new building in May, 2006. This outstanding building, which meets the LEED Gold Standard, contains lecture theatres on the first floor; the remaining five floors are devoted to offices, undergraduate laboratories, research laboratories and all associated space for the Department of Computer Science. Since we are essentially the sole occupants of the building, we were able to design the space almost exactly as we required for our needs. This has resulted in state of the art laboratories, and excellent facilities for the department. We are particularly pleased that all offices and labs in the building (except for a single small office) have natural light. The new building has been of inestimable value in building the morale of the department, as we were reunited into a single location after being split across campus for some years.

During the last three years, the department underwent two moves: The first one was in 2004 when all research labs and graduate student offices were moved to TEF; the second was in May 2006 when the whole department moved to the new ECS building. The historical data is shown in Appendix A, Table A.1. During the transition, we were able to increase our research space in 2004 from 6,262 square feet to 7,648 square feet and then more than double our research space in 2006 to 17,630 square feet. The space allocated for undergraduate students, including teaching laboratories, teaching support, and group meeting rooms, was also increased from 7,766 square feet to 9,682 square feet in 2006.

It is unusual for a department to undergo two transitions during such a short period and we did experience many difficulties because of the moves. Both moves required substantial efforts in packing, unpacking, and setting up research equipment. This was a very time-consuming task and could not be done without effective scheduling and coordination. It is worthwhile noting that the first move in 2004 resulted in the painful separation of graduate students offices from faculty offices. For nearly two years, graduate students felt isolated from the department, and such an inconvenience had a negative impact on our research. Nevertheless, thanks to the great efforts of our administrative and technical support staff, we have overcome all difficulties during the moves. We are very proud that we were able to smoothly transfer between the different locations, and we feel very relieved that we have finally settled down without worrying about another move for a long time.

The historical data plus the current data, as shown in Tables 7.4 to 7.7, illustrates that we are now in good shape regarding space allocation. Adequate space, however, has been a long-term challenge for the department. We have been faced with three major difficulties during the last several years. First, the number of our regular faculty members has nearly doubled, resulting in a big expansion of several research areas such as Software Engineering, Computer Networks, Theory, and Computer Graphics. Such research expansion makes the pressure for research space relentlessly high in our department. Second, six faculty members have held, and more new faculty members are applying for, CFI funds for research infrastructure. Research space is required for all CFI projects. Third, our department is responsible for most computer-related service courses to the whole university. To this end, our teaching lab facilities are open to all students taking computer courses in our department, leading to heavy pressure on our teaching laboratory space and related supporting resources.

With the current department size, the average space per researcher (including regular faculty members and graduate students) in our department is about 120 square feet, which is not spacious but definitely enough for performing research. Unlike the research space, the space for undergraduate teaching has increased only slightly during the last several years. We are moving in a positive direction but the pressure on undergraduate teaching space still exists. Looking ahead, with the industry recovering from a winter season and with the hard work of our outreach program committees, we expect that our student enrollment will increase significantly and our space will be “full” again. We may need to find ways for further expansion in the next five years.

TABLE 7.4: Statistical Data on Space Designation

		Before 2003 (ft^2)	2004-2006 (ft^2)	Since May 2006 (ft^2)
Research Space	Research Laboratories & Graduate Students/ Visitors Offices	6,262	7,648	16,563
	Faculty of Engineering Space (meeting rooms and lounges)	1,949 (shared)	599 1,949 (shared)	1,067 2,045 (shared)
Undergraduate teaching laboratories & support, group study rooms		7,766	7,766	9,682
Total Size		14,028 1,949 (shared)	16,013 1,949 (shared)	27,312 2,045 (shared)

TABLE 7.5: Space Information (Since May 2006)

		Rooms	Total (ft^2)
Research Space	Research Laboratories & Graduate Students/ Visitors Offices	ECS: 302, 304, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 330, 402, 404, 405, 407, 411, 412, 413, 415, 418, 419, 420, 421, 422, 423, 424, 430, 431, 442, 448, 455, 457, 459, 461, 463, 465, 461, 515, 517, 519, 521, 523, 527, 542, 548, 555, 557, 559, 561, 563, 565, 567, 602, 604, 607, 609, 611, 613, 615, 642, 648, 654	16,563
	Faculty of Engineering Space (meeting rooms and lounges)	ECS: 531, 668 Shared space: 405, 407, 467, 468, 660, 223	1,067 2,045 (shared)
Undergraduate teaching laboratories & support, group study rooms		ECS: 242, 249, 250, 251, 253, 255, 257, 258, 266, 321, 322, 323, 324, 325, 326, 328, 331, 342, 348, 349, 354, 360, 366, 423	9,682

TABLE 7.6: Space Information (Between 2004 and 2006)

		Rooms	Total (ft^2)
Research Space	Research Laboratories & Graduate Students/ Visitors Offices	TEF: 222, 220, 218, 241, 243, 245, 240, 242, 244, 246, 248, 250, 252, 254, 256, 258, 253A, 230, 232, 234, 236, 253, 255, 257, 259, 231, 233, 235, 237, 238	7,648
	Faculty of Engineering Space (meeting rooms and lounges)	TEF: 225, 247 Shared space: EOW: 330, 230, 430, 502, ELWA 227	599 1,949 (shared)
Undergraduate teaching laboratories & support, study rooms		ELW 207, A209, A243, B203, B215, B220, B228, B210,	7,766

TABLE 7.7: Space Information (Before 2003)

		Rooms	Total (ft^2)
Research Space	Research Laboratories & Graduate Students/ Visitors Offices	ELWA: 104, 232, 234, 236, 238, 240, 242, 246, 248, 250, 254, 328, 330, 334, 342, 344, 213, 219, 221, 228, 229, 233, 237 ELW 306	6,262
	Faculty of Engineering Space (meeting rooms and lounges)	Shared space: EOW 330, 230, 430, 502 ELWA 227	1,949 (shared)
Undergraduate teaching laboratories & support		ELW 207, A209, A243, B203, B215, B220, B228, B210	7,766

Appendix A: The North American Context

We include here some comparative data for Computer Science departments in Canada and North America with Computer Science at Victoria. The data comes from the annual CRA Taulbee Report and the annual CAUT Almanac.

A.1 Faculty

Table A.1 shows the numbers for faculty in Computer Science at UVic and in Canada overall, divided by gender, with the corresponding percentages. Table A.2 shows faculty demographics trends at UVic and in Canada.

TABLE A.1: Faculty members in CSc: UVic and Canada

	UVic			Canada			Percentage: UVic/ Canada		
Years	**M**	**F**	**Total**	**M**	**F**	**Total**	**M**	**F**	**Total**
2001/02	18	6	24	695	129	824	2.59%	4.65%	2.91%
2002/03	20	7	27	748	148	932	2.67%	4.73%	2.90%
2003/04	23	8	31	888	168	1056	2.59%	4.76%	2.93%
2004/05	23	8	31	918	183	1101	2.51%	4.27%	2.82%

TABLE A.2: Faculty demographics in CSc: UVic and Canada

Age	CSc at UVic	CSc in Canada
40 or less	32.4%	33.4%
41 - 50	23.6%	32.3%
51 - 60	32.4%	24.1%
Over 60	11.8%	10.1%

A.2 Undergraduate Students

The number of CSc degrees awarded can be compared to the overall numbers in North America, together with corresponding percentages, as shown in Table A.3. The number of degrees awarded in North America in 2005/06 is down by 27.7% from the peak; at the University of Victoria, the number is down by 21% from the peak.

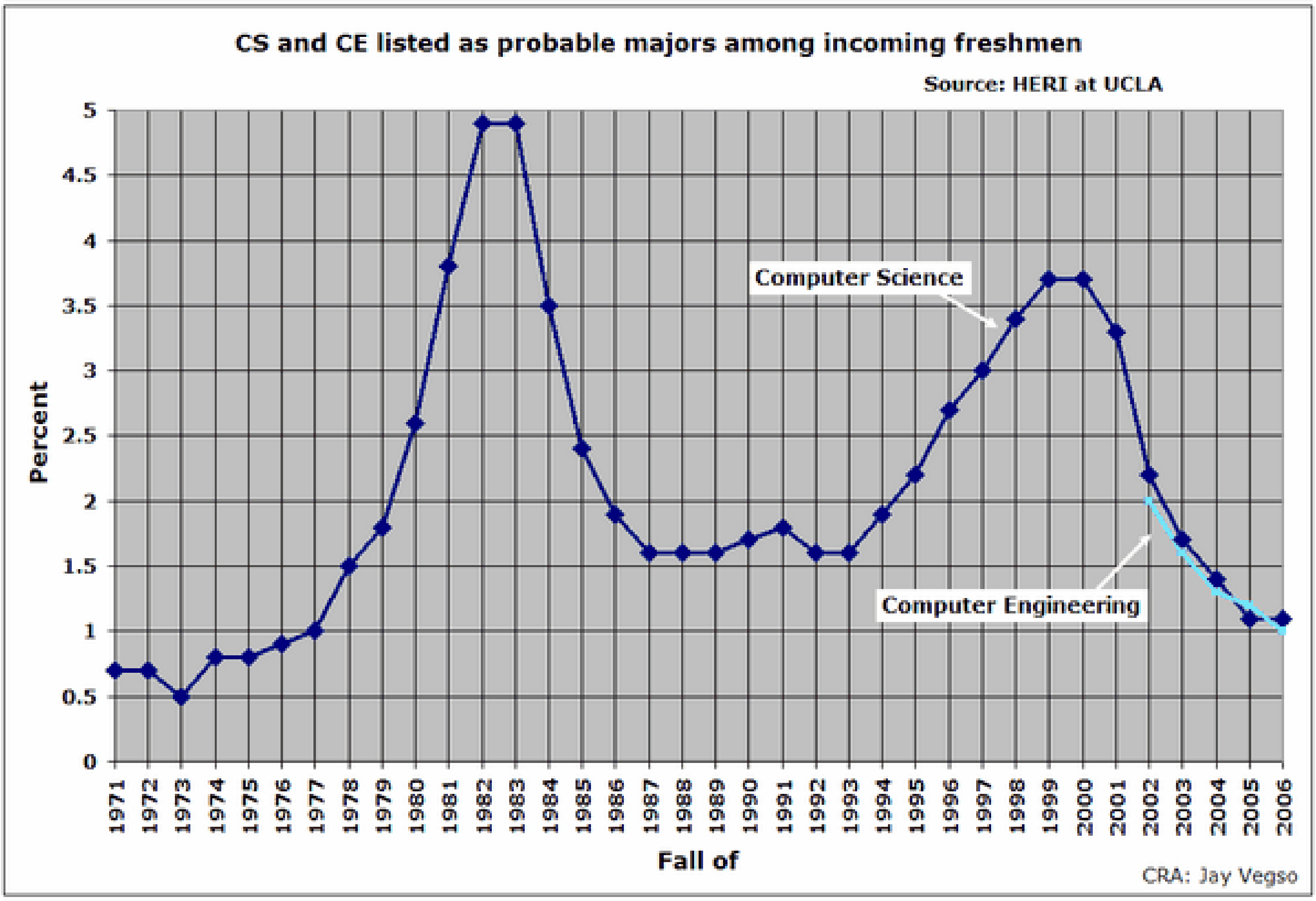
Attempts to measure and compare the headcounts in programs are very difficult because of the uncertainty of the measure. Some places admit students in year 1, while in other places students do not have to declare until the third year (as it is here mainly). Further, the comparisons between years are problematic as more institutions are currently changing their admission towards year 1. However, the enrolment both across Canada and at UVic appears to be down by about 40% from the peak.

TABLE A.3: CSc Degrees: UVic and North America

Years	UVic	North America	Percentage: UVic/North America
1998/99	83	10,809	0.77%
1999/00	125	12,660	0.99%
2000/01	121	14,427	0.84%
2001/02	135	16,907	0.80%
2002/03	157	16,633	0.94%
2003/04	153	17,876	0.86%
2004/05	170	15,137	1.12%
2005/06	124	12,929	0.96%

More problematical, the number of incoming students choosing computer science is also falling rapidly, as evidenced from the study undertaken by the Higher Education Research Institute at the University of California at Los Angeles (HERI/UCLA) based on surveys at 131 institutions in the U.S.A. (see Figure A.1). After peaking in 1999 and 2000, interest in CSc as a major fell 70% between 2000 and 2005. In the fall of 2006, 1.1% of incoming freshmen chose CSc as their probable major, the same as in 2005.

FIGURE A.1. HERI/UCLA Study



A.3 Graduate Students

Across North America, the number of masters students is down by 20% from the peak in 2003/04. That is, the decline has taken place over just two years. Canada has not, as yet, seen anything like a similar decline. Table A.4 shows the current data.

TABLE A.4: Enrolment in Masters and Doctoral Programs

Years	UVic	Canada	Percentage: UVic/ Canada
2001/02	89	2520	3.53%
2002/03	101	2695	3.74%
2003/04	114	3354	3.40%
2004/05	119	3872	3.07%
2005/06	119	3356	3.55%

Appendix B: Publications by Regular Faculty (2000–2007)

The publications listed in this Appendix have been extracted from the CVs on file with the department as of 21 March 2007. It includes the fully refereed journal publications listed in section 7b and the fully refereed conference publications listed in section 7c. To maintain consistency, any conference publication listed as a poster presentation has been omitted.

Publications in the range 2000 to the present have been included, except that publications which predate a faculty member's appointment at UVic are omitted. The list includes professors emeritus up to the dates of their retirement and includes senior instructors.

B.1 Refereed Journal Publications 2000-2007

Akbar, M.M., M.S. Rahman, M. Kaykobad, E.G. Manning, and G.C. Shoja, "Solving the multidimensional multiple-choice knapsack problem by constructing convex hull," Computers & Operations Research, accepted Oct. 2004.

Alber, J., H. Fan, M. Fellows, H. Fernau, R. Niedermeier, F. Rosamond, and U. Stege, "Refined search tree techniques for the planar dominating set problem," accepted to Journal of Computer and Systems Sciences, vol. 71, no. 4, pp. 385-405, 2005.

Ardis, M., N. Daley, D. Hoffman, H. Siy and D. Weiss, "Software product lines: a case study," Software - Practice and Experience, vol. 30, no. 7, May 2000.

Aycock, J., and R.N. Horspool. "Practical Earley Parsing," Computer Journal, vol. 45, 6, pp. 620-630, Nov. 2002.

Aycock, J., and R.N. Horspool, J. Janousek and B. Melichar. "Even faster generalized L.R. Parsing," Acta Informatica, vol. 37, 9, pp. 633-651, Sep. 2001.

Aycock, J., and R.N. Horspool. "Schrödinger's token," Software - Practice and Experience, vol. 31, 8, pp. 803-814, July 2001.

Ball, T., D. Hoffman, F. Ruskey, R. Webber, and L. White, "State generation and automated class testing: a case study," Software Testing Verification and Reliability, 10, pp. 149-170, 2000.

Bingham, B.D., D.D. Olesky, and P. van den Driessche, "Potentially nilpotent and spectrally arbitrary even cycle sign patterns," Linear Algebra and its Applications, 421, pp. 24-44, 2007.

Bishop, J., and R.N. Horspool, "Cross-platform development: software that lasts, IEEE Computer, pp. 26-35, Oct. 2006.

Bishop, J., R.N. Horspool, and B. Worrall, "Experience in integrating Java with C# and NET," Concurrency and Computation: Practice and Experience, 17, 5-6, pp. 663-680, Apr.-May 2005.

Boyer, J., and W. Myrvold, "On the cutting edge: simplified O(n) planarity by edge addition," Journal of Graph Algorithms and Applications, vol. 8, no. 3, pp. 241-273, 2004.

Britz, T., D.D. Olesky, and P. van den Driessche, "Matrix inversion and digraphs: the one factor case," Electronic Journal of Linear Algebra, 11, pp. 115-131, 2004.

Britz, T., D.D. Olesky, and P. van den Driessche, "Schur complements of matrices with acyclic bipartite graphs," Electronic Journal of Linear Algebra, 14, pp. 2-11, 2005.

Britz, T., D.D. Olesky, and P. van den Driessche, "The Moore-Penrose inverse of matrices with an acyclic bipartite graph," Linear Algebra and its Applications, 390, pp. 47-60, 2004.

Britz, T., J.J. McDonald, D.D. Olesky, and P. van den Driessche, "Minimal spectrally arbitrary sign patterns," SIAM Journal on Matrix Analysis and Applications, 26, pp. 257-271, 2004.

Buss, S.R., and B.M. Kapron, “Resource bounded continuity and sequentiality for type-two functionals,” ACM Transactions on Computational Logic, vol. 3, pp. 402-417, 2002.

Cai, L., J. Ye, J. Pan, X. Shen, and J. Mark, “Dynamic server selection using fuzzy inference in content distribution networks,” Elsevier Journal of Computer Communications, vol.29, no. 8, pp. 1026-1038, 2006.

Cai, L., X. Shen, J. Mark, and J. Pan, “QoS support for multimedia traffic in wireless/wireline networks using TCP-friendly AIMD protocols,” IEEE Transactions on Wireless Communications, vol. 5 no. 2, Feb. 2006.

Cai, L., X. Shen, J. Pan, and J. Mark, “Performance analysis of TCP-friendly AIMD algorithms for multimedia applications,” IEEE Transactions on Multimedia, vol. 7, no. 2, pp. 339-355, 2005.

Cai, L., X. Shen, J. W. Mark, and J. Pan, “Performance modeling and analysis of window-controlled multimedia flows in wireless/wired networks,” IEEE Transactions on Wireless Communications, accepted, 2006.

Cai, M., K. Hwang, J. Pan, and C. Papadopoulos, “WormShield: Fast Worm Signature Generation with Distributed Fingerprint Aggregation,” IEEE Transactions on Dependable and Secure Computing, accepted, 2006.

Carroll, J., F. Ruskey, and M. Weston, “Which n-Venn diagrams can be drawn with convex k-gons?” Discrete and Computational Geometry, to appear.

Cattell, K., F. Ruskey, C.R. Miers, J. Sawada, and M. Serra, “The number of irreducible polynomials over GF(2) with given trace and subtrace,” Journal of Combinatorial Mathematics and Combinatorial Computing, 34 pages, 47, pp. 31-64, 2003.

Cattell, R., F. Ruskey, J. Sawada, C.R. Miers, and M. Serra, “Generating Unlabelled Necklaces and Irreducible Polynomials over GF(2),” Journal of Algorithms, vol. 37, no. 2, pp. 267-282, Nov. 2000.

Chambers, J., A. Gagarin, and W. Myrvold, “The obstructions for toroidal graphs with no K3,3's,” Discrete Mathematics, 10 pages, accepted Aug. 2006.

Chandran, L.S., L. Ibarra, F. Ruskey, and J. Sawada, “Fast generation of all perfect elimination orderings of a chordal graph,” Theoretical Computer Science, 307, pp. 303-317, 2003.

Cheetham, J., F. Dehne, A. Rau-Chaplin, U. Stege, and P.J. Taillon, “Solving large FPT problems on course grained parallel machines,” Journal of Computer and Systems Sciences, vol. 67, no. 4, pp. 691-706, Dec. 2003.

Cheung, Wai-Shun, Chi-Kwong Li, D.D. Olesky and P. van den Driessche, “Optimizing quadratic forms of adjacency matrices of trees and related eigenvalue problems,” Linear Algebra and its Applications, 325, pp. 191-207, 2001.

Chow, S., and F. Ruskey, “Drawing minimum area Venn Diagrams using polyominoes,” Mathematics Magazine (MAA), to appear in April issue, 13 journal pages.

Christl, A., R. Koschke, and M.-A. Storey, “Automated clustering to support the reflexion method,” Information and Software Technology, Elsevier, 2006.

Colbourn, C., D. Drake, and W. Myrvold, “Ovals and hyperovals in nets,” Discrete Mathematics, vol. 294, no. 1-2, pp. 53-74, 2005.

Cooper, M., J. Foote, E. Pampalk, and G. Tzanetakis, “Visualization in audio-based music information retrieval,” Computer Music Journal, 30(2), 2006.

Daley, N., D. Hoffman and P.Strooper, “A framework for table driven testing of Java classes,” Software-Practice and Experience, vol. 32, no. 5, April 2002.

Dame, L.F., D.D. Olesky, and P. van den Driessche, “The exponent and circumdiameter of primitive digraphs,” Linear Algebra and its Applications, 396, pp. 243-258, 2005.

Damian Herlea, D.E., A. Eberlein, M.L.G. Shaw, and B.R. Gaines, “Using different communication media in requirements negotiation,” IEEE Software, Best Paper Award at IEEE Int. Conf. on Requirements Engineering (ICRE 2000), pp. 28-36, May/June 2000.

Damian, D., “Stakeholders in global requirements engineering: lessons learned from practice,” IEEE Software, Mar./Apr. 2007, to appear.

Damian, D., and J. Chisan, “An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality and risk management,” Journal of Transactions in Software Engineering, 32(7), pp. 433-453, 2006.

Damian, D., D. Zowghi, “Requirements challenges in multi-site software development organizations,” Requirements Engineering Journal 8, pp. 149-160, 2003.

Damian, D., D. Zowghi, L. Vaidyanathasamy, and Y. Pal, “An industrial case study of immediate benefits of requirements engineering process improvement at the Australian Center for Unisys Software,” Int. Journal of Empirical Software Engineering, 9(1-2), pp. 45-75, Mar. 2004.

Damian, D., L. Vaidyanathasamy, and Y. Pal, “Requirements Engineering and Downstream Software Development: Findings from a case study,” Journal of Empirical Software Engineering, 10(3), 2005.

Damian, D.E.H., “A multimedia meeting system in support of distributed requirements negotiations,” e-Service Journal, Special Issue on Electronic Collaboration, 1(3), pp. 43-60, 2002.

Damian, D.E.H., A. Eberlein, M.L.G. Shaw, and B.R. Gaines, “Facilitation in distributed requirements engineering,” Requirements Engineering Journal, 2001.

Dannenberg, R., et al., “A comparative evaluation of search techniques for query-by-humming using the Musart testbed,” Journal of American Society for Information Science and Technology, (58(3), scheduled for Feb. 2007.

Dannenberg, R., W. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo, “The MUSART testbed for query-by-humming evaluation,” in Computer Music Journal (Special Issue on Music Information Retrieval), 28 (2), pp. 34-48, 2004.

Daugherty, S., W. Myrvold, and P.W. Fowler, “Backtracking to compute the independence number of a fullerene,” accepted to MATCH, Oct. 2006.

Debroni, B., S. Girn, M. Minchenko, and J. Woodcock, “FUIGUI: a graphical user interface for investigating conjectures about fullerenes,” accepted to MATCH, Nov. 2006.

Drake, D., and W. Myrvold, “Nets of small degree without ovals,” Designs, Codes and Cryptography, vol. 32, no. 1-3, pp. 167-183, 2004.

Drake, D., and W. Myrvold, “The non-existence of maximal sets of four mutually orthogonal Latin squares of order 8,” Designs, Codes and Cryptography, vol. 33, no. 1, pp. 63-69, 2004.

Drew, J.H., C.R. Johnson, D.D. Olesky and P. van den Driessche, “Spectrally arbitrary patterns,” Linear Algebra and its Applications, 308, pp. 121-137, 2000.

Dubrova, E.V., and J.C. Muzio, “Easily testable multiple-valued logic circuits derived from Reed-Muller circuits,” IEEE Trans. Computers, vol. 49, pp. 1285-1289, 2000.

Effler, S., and F. Ruskey, “A CAT algorithm for generating permutations with a fixed number of inversions,” Information Processing Letters, 86, pp. 107-112, 2003.

El-Kharashi, W.M., G. Darling, Brent Marykuca, and G.C. Shoja, “Understanding and implementing computer network protocols through a lab project,” IEEE Trans. on Education, vol. 45, no. 3, pp. 276-284, Aug. 2002.

Ellis, J., F. Ruskey, J. Sawada, and J. Simpson, “Euclidean strings,” Theoretical Computer Science, 301, pp. 321-340, 2003.

Elsner, L., D.D. Olesky, and P. van den Driessche, “Low rank perturbations and the spectrum of a tridiagonal sign pattern,” Linear Algebra and its Applications, 374, pp. 219-230, 2003.

Ernst, N., M.-A. Storey, and P. Allen, “Cognitive support for ontology engineering,” Int. Journal of Human-Computer Studies, vol. 62, no. 5, pp. 553-577, May 2005.

Fellows, M.R., M.T. Hallett, and U. Stege, “Analogs and duals of the MAST problem for sequences and trees,” Journal of Algorithms, vol. 49, no. 1, pp. 192-216, 2003.

Fowler, P.W., D. Horspool, and W. Myrvold, “Vertex spirals in fullerenes and their implications for nomenclature of fullerene derivatives,” Chemistry: A European Journal, vol. 13, no. 8, pp. 2208-2217, 2007.

Galanaki, C., Rondogiannis, P., and W.W. Wadge, “An infinite-game semantics for well-founded negation in logic programming,” accepted to Annals of Pure and Applied Logic (special issue on game semantics).

German, D.M., "An empirical study of fine-grained software modifications," Journal of Empirical Software Engineering, vol. 16, no. 1, pp. 5-22, Feb. 2006.

German, D.M., "Decentralized open source global software development, the GNOME experience," Journal of Software Process: Improvement and Practice, vol. 8, no.4, pp. 201-215, 2004.

German, D.M., "Using software trails to reconstruct the evolution of software," Journal of Software Maintenance and Evolution: Research and Practice, vol. 16, no. 6, pp. 367-384, 2004.

German, D.M., and A. Hindle, "Visualizing the evolution of software using softChange," in Journal of Software Engineering Knowledge Engineering, Special Issue of Best Papers SEKE 2004, vol.16, no. 1, pp. 4-22, Feb. 2006.

German, D.M., and A. Hindle, "Visualizing the evolution of software using softChange," Journal of Software Engineering Knowledge Engineering.

Gibbs, C., Y. Coady, M. Haupt, J. Vitek and H. Yamauchi, "A domain specific aspect language for virtual machines," IEE Proceedings Software, (to appear).

Goranko, V., and B.M. Kapron, "The modal logic of the countable random frame," Archive for Mathematical Logic, vol. 42, pp. 221-243, 2003.

Gouyon, F., et al, "An experimental comparison of audio tempo induction algorithms," IEEE Transactions of Audio, Speech and Language Processing, 14(5), Sept. 2006.

Grahne, G., and A. Thomo, "Regular path queries under approximate semantics," Annals of Mathematics and Artificial Intelligence, Springer, vol. 46, no. 1-2, pp. 165-190, 2006.

Harsha, P., Y. Ishai, J. Kilian, K. Nissim, and S. Venkatesh, "Communication vs. Computation," Computational Complexity, to appear.

Hastad, J., and S. Venkatesh, "On the advantage over a random assignment," Random Structures and Algorithms, 25(2), pp. 117-149, 2004.

Herlea, D.E., C.M. Jonker, J. Treur, N.J.E. Wijngaards, "A compositional knowledge level process model of requirements engineering," Int. Journal of Software Engineering and Knowledge Engineering, 12(1), pp. 41-75, 2002.

Higham, L., L. Jackson, and J. Kawash, "Specifying memory consistency of write buffer multiprocessors," ACM Transactions on Computer Systems, to appear.

Higham, L., L. Jackson, and J. Kawash, "What is Itanium memory consistency from the programmers point of view?" Electronic Notes in Theoretical Computer Science, to appear.

Hoffman, D., and P. Strooper, "API documentation with executable examples," Journal of Systems and Software, vol. 66, no. 2, pp. 143-156, 2003.

Hoffman, D., P. Strooper, and S. Wilkin, "Tool support for executable documentation of Java class hierarchies," Journal of Software Testing, Verification and Reliability, vol. 15, no. 1, April 2005.

Hou, Y., and J. Pan, "Analysis and evaluation of expiration-based hierarchical caching systems," Elsevier Journal of Performance Evaluation, vol. 55, nos. 1-2, pp. 51-68, 2004.

Hou, Y., Y. Shi, J. Pa, and S. Midkiff, "Maximizing lifetime of wireless sensor networks through optimal single-session flow routing," IEEE Transactions on Mobile Computing, vol. 5, no. 9, pp. 1255-1266, 2006.

Hou, Y., J. Pan, B. L, and S. Panwar, "On expiration-based hierarchical caching systems," IEEE Journal on Selected Areas in Communications (JSAC), vol. 22, no.1. pp. 134-150, 2004.

Impagliazzo, R., and B.M. Kapron, "Logics for reasoning about cryptographic constructions," Journal of Computer and System Sciences, vol. 76, pp. 286-320, 2006.

Irwin, R., B.M. Kapron, and J. Royer, "Characterizations of the basic feasible functionals, part I," Journal of Functional Programming, vol. 11, pp. 117-153, 2001.

Jackson, B., and F. Ruskey, "Meta-fibonacci sequences, binary trees, and extremal compact codes," Electronic Journal of Combinatorics, vol. 13, no. R26, 13 pages, 2006.

Jahnke, J. H., Y. Bychkov, D. Dahlem, and L. Kawasme, "Semantic context-based composition of portal components," Journal of Organizational and End User Computing (JOEUC), vol. 18, issue 4, pp. 66-87, Idea Group, 2006.

Jahnke, J.H. , W. Schäfer, S. Wadsack and A. Zundorf, “Supporting iterations in exploratory database reengineering processes,” Science of Computer Programming, Elsevier Publishers, vol. 45, no. 2-3, pp. 99-136, 2002.

Jahnke, J.H., “Cognitive support in software reengineering based on generic fuzzy reasoning nets,” Fuzzy Sets and Systems, vol. 145, no. 1, pp. 3-27, July 1, 2004.

Jahnke, J.H., A. Walenstein, “Evaluating theories for managing imperfect knowledge in database reengineering processes,” Int. Journal of Software Engineering and Knowledge Engineering (IJSEIKE), 12 (1), pp. 77-102, Feb. 2002.

Jahnke, J.H., M. d'Entremont, and J. Stier, “Facilitating the programming of the smart home,” IEEE Wireless Communications, 9 (6), pp. 70-76, Dec. 2002.

Jahnke, J.H., Y. Bychkov, D. Dahlem, L. Kawasme, “Context-aware information delivery,” Revue d'intelligence artificielle, vol. 19, no. 3, Lavoisier, 2005.

Johnson Jr., P.D., W. Myrvold, and K.J. Roblee, “More extremal problems for edge-regular graphs,” accepted to Utilitas Mathematica, 10 pages, 2005.

Johnson, C.R. and D.D. Olesky, “Rectangular submatrices of inverse M-matrices and the decomposition of a positive matrix as a sum,” Linear Algebra and its Applications, 409, pp. 87-99, 2005.

Johnson, C.R. and D.D. Olesky, “Sums of totally positive matrices,” Linear Algebra and its Applications, 392, pp. 1-9, 2004.

Johnson, C.R., D.D. Olesky and P. van den Driessche, “Successively-ordered elementary bidiagonal factorization,” SIAM Journal on Matrix Analysis and Applications, 22, pp. 1079-1088, 2001.

Johnson, C.R., D.D. Olesky, and P. van den Driessche, “Matrix classes that generate all matrices with positive determinant,” SIAM Journal on Matrix Analysis and Applications, 25, pp. 285-294, 2003.

Kent, K.B., M. Serra, and R.N. Horspool, “Hardware/software co-design for virtual machines,” IEE Computer & Digital Techniques, 152, 5, pp. 537-696, Sept. 2005.

Khan, S., E.G. Manning and G. Shoja, “Optimal quality of service routing and admission control using the utility model,” Fifth Generation Computer Systems, Feb. 2003.

Khan, S., K.F. Li, E.G. Manning, R. Watson, and G.C. Shoja, “Optimal quality of service routing and admission control using the utility model,” accepted for publication in Future Generation Computer Systems (FGCS) Journals, Feb. 14, 2003.

Khan, S., K.F. Li, E.G. Manning and M. Akbar, “Solving the knapsack problem for adaptive multimedia systems,” with Studia Informatica Special Issue on Combinatorial Problems, vol. 2, no. 1, 2002.

Kienle, H.H., D.M. German, S. Tilley, and H. Müller, “Managing legal Risks associated with intellectual property on the web,” Int. Journal of Business Information Systems, accepted for publication, vol. 3, no. 1, 2008.

Killian, C., F. Ruskey, C. Savage, and M. Weston, “Half-simple symmetric Venn diagrams,” Electronic Journal of Combinatorics, vol. 11, no. R86, 22 pages, 2004.

Kim, I.-J., D.D. Olesky, and P. van den Driessche, “Inertially arbitrary sign patterns with no nilpotent realization,” Linear Algebra and its Applications, 421, pp. 264-283, 2007.

Kim, I.-J., J.J. McDonald, D.D. Olesky and P. van den Driessche, “Inertias of zero-nonzero patterns,” Linear and Multilinear Algebra, 55, pp. 229-238, 2007.

King, V., and G. Sagert, “A fully dynamic algorithm for maintaining the transitive closure,” Journal of Computer and System Sciences, 65, pp. 150-167, 2002.

King, V., and M.R. Henzinger, “Maintaining minimum spanning trees in dynamic graphs,” SIAM Journal on Computing, vol. 31, no. 2, pp. 364-374, 2001.

Kingan, S.R.J., and W. Myrvold, “On Matroid generation,” Proc. of 34th Southeastern Int. Conf. on Combinatorics, Graph Theory and Computing, Congressus Numerantium, vol. 164, pp. 95-109, 2003.

Kirkland, S., D.D. Olesky and P. van den Driessche, “Digraphs with large exponent,” The Electronic Journal of Linear Algebra, 7, pp. 30-40, 2000.

Krall, A., Y. Pryanishnikov, and R.N. Horspool, “Compiler optimizations for processors with SIMD instructions, Software - Practice and Experience, vol. 37, no.1 (Jan.), p.p. 93-113, Jan. 2007.

Kube, N., and F. Ruskey, "Sequences that satisfy a(n-a(n))=0," Journal of Integer Sequences, article 05.5.5, vol. 8, 8 pages, 2005.

Kuziemsky, C., J. H. Jahnke, and F. Lau, "The e-hospice - beyond traditional boundaries of palliative care," Telematics and Informatics, vol. 23, no. 2, pp. 117-, Elsevier, 2006.

Kuziemsky, C., and J. H. Jahnke, "A multimodal approach to health care decision support systems, " Communications of the Association for Information Systems, vol. 16, no. 19, 2005.

Layman, L., L. Williams, D. Damian, and H. Bures, "Essential communication practices for extreme programming in a global software development team," Journal of Software and Technology, to appear.

Li, R., J. Li, K. Wu, Y. Xiao, and L. Xie, "An enhanced fast handover scheme for mobile IPv6," accepted by IEEE Transactions on Wireless Communications, 2006.

Lin, Q., D.D. Olesky, and P. van den Driessche, "The distance of potentially stable sign patterns to the unstable matrices," SIAM Journal on Matrix Analysis and Applications, 24, pp. 356-367, 2002.

Liu, C., K. Wu, and J. Pei, "An energy efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," accepted by IEEE Transactions on Parallel and Distributed Systems (TPDS), 2006.

Liu, C., K. Wu, Y. Xiao, B. Sun, "Random coverage with guaranteed connectivity: joint scheduling for wireless sensor networks," IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 17, no. 6, pp. 562-575, June 2006.

Liu, C., T. Scott, K. Wu, and D. Hoffman, "Range-free sensor localization with ring overlapping based on comparison of received signal strength indicator," accepted by Int. Journal of Sensor Networks, 2007.

Liu, R., C. Gibbs, and Y. Coady, "Safe and sound evolution with SONAR: sustainable optimization and navigation with aspects for system-wide reconciliation," Transactions on Aspect-Oriented Software Development, Springer; special Issue on Aspect-Oriented Programming and Development for Software Evolution (to appear).

Long, B., D. Hoffman, and P. Strooper, "Tool support for testing concurrent Java components," IEEE Trans. Soft. Eng., vol. 29, no. 6, pp. 555-566, June 2003.

Mark, J., J. Pan, and X. Shen, "Mobility support in hybrid wireless/IP networking," Elsevier Journal of Computer Communications, vol. 26, no. 17, pp. 1990-1997, 2003.

McDonald, J.J., D.D. Olesky, M.J. Tsatsomeros, and P. van den Driessche, "On the spectra of striped sign patterns," Linear and Multilinear Algebra, 51, pp. 39-48, 2003.

McKay, B.D., W. Myrvold, and A. Meynert, "Small Latin squares, quasigroups and loops," Journal of Combinatorial Designs, vol. 15, no. 2, pp. 98-119, 2006.

Maslov, D., G. Dueck, and D. M. Miller, "Toffoli Network Synthesis with Templates," IEEE Transactions on Computer Aided Design, vol. 24, issue 6, pp. 807-817, June 2005.

Maslov, D., G. Dueck, and D. M. Miller, "Two-step Synthesis of Toffoli-Fredkin Reversible Networks," IEEE Transactions on Very Large Scale Integration Systems, vol. 13, issue 6, pp. 765-769, June 2005.

Maslov, D. and D. M. Miller, "Comparison of the Cost Metrics through Investigation of the Relation between Optimal NCV and Optimal NCT 3-qubit Reversible Circuits," IET Computers & Digital Techniques, accepted.

Miers, C.R., and F. Ruskey, "Counting strings with given elementary symmetric function evaluations I: strings over Zp with p prime," SIAM Journal of Discrete Mathematics, vol.17, pp. 675-685, 2004.

Miers, C.R., and F. Ruskey, "Counting strings with given elementary symmetric function evaluations II: circular strings," SIAM J. Discrete Mathematics, vol, 18 pp. 71-82, 2004.

Miers, C.R., F. Ruskey, and J. Sawada, "Generating Lyndon brackets: a basis for the n-th homogeneous component of the free Lie algebra," Journal of Algorithms, 46 (2003) pp. 21-26.

Miller, M., J. Gimbert, F. Ruskey, and J. Ryan, "Iterations of eccentric digraphs," Bulletin of the Institute of Combinatorics and Its Applications, vol. 45, pp. 41-50, 2005.

Miller, D.M., D. Maslov, and G. Dueck, "Synthesis of Quantum Multiple-Valued Circuits," Journal of Multiple-Valued Logic and Soft Computing, vol. 12, no. 5-6, pp. 431-450, 2006.

Myrvold, W., and F. Ruskey, "Ranking and unranking permutations in linear time," Information Processing Letters, vol. 79, pp. 281-284, 2001.

Myrvold, W., and J. Roth, “Simpler projective plane embedding,” Ars Combinatoria, vol. 75, pp. 135-155, 2005.

Myrvold, W., T. Prsa, and N. Walker, “A dynamic programming approach for timing and designing clique algorithms,” in DIMACS Series in Discrete and Theoretical Computer Science, S. Fajtlowicz, P.W. Fowler, P. Hansen, M. Janowitz, and F.S. Roberts (editors), vol. 69, pp. 333-340, 2005.

Olesky, D.D., Aidan Roy, and P. van den Driessche, “Maximal graphs and graphs with maximal spectral radius,” Linear Algebra and its Applications, 346, pp. 109-130, 2002.

Olesky, D.D., B. Shader, and P. van den Driessche, “ Permanents of Hessenberg (0,1)-matrices,” Electronic Journal of Combinatorics, 12, R70, 25 pp., 2005.

Olesky, D.D., Bryan Shader, and P. van den Driessche, “Exponents of tuples of nonnegative matrices,” Linear Algebra and its Applications, 356, pp. 123-134, 2002.

Olesky, D.D., P. van den Driessche and J.H. Verner, “Graphs with the same determinant as a complete graph,” Linear Algebra and its Applications, 312, pp. 191-195, 2000.

Pan, J., L. Cai, Y. Hou, Y. Shi, and X. Shen, “Optimal base-station locations in two-tiered wireless sensor networks,” IEEE Transactions on Mobile Computing, vol. 6, no. 5, pp. 458-473, 2005.

Pan, J., Y. Hou, and B. Li, “An overview of DNS-based server selections in content distribution networks,” Elsevier Journal of Computer Networks, vol. 43, no. 6, pp. 695-711, 2003.

Pan, J., J. Mark, and X. Shen, “TCP performance and behaviors with local retransmissions,” Kluwer Journal of Super Computing, vol. 23, no. 3, pp. 225-244, 2002.

Pan, J., J. Mark, and X. Shen, “TCP performance with segment-in-flight estimation algorithm over wireless links,” Watam Journal of Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications and Algorithms, vol. 13, no. 1, pp. 77-94, 2006.

Pan, J., L. Cai, and X. Shen, “Vulnerabilities in distance-indexed IP traceback schemes,” Int. Journal of Security and Networks, in press, 2006.

Pan, J., Y. Hou, and B. Li, “Freshness and retrieval thresholds in expiration-based hierarchical caching systems,” Elsevier Journal of Computer Networks, vol. 44, no. 2, pp. 235-246, 2003.

Pan, J., Y. Hou, L. Cai, Y. Shi, and X. Shen, “Serialized optimal relay schedules in two-tiered wireless sensor networks,” Elsevier Journal of Computer Communications, vol. 29, no. 4, pp. 511-524, 2006.

Pan, L., L. Cai, X. Shen, and J. Mark, “Identity-based secure collaboration in wireless ad hoc networks,” Elsevier Journal of Computer Networks, vol. 51, no. 3, pp. 853-865, 2007.

Pang, J.C., G.C. Shoja, and E.G. Manning, “Providing soft real-time QoS guarantees for java threads,” in Concurrency and Computation Practice and Experience, vol. 15, nos. 3-5, pp. 521-538, March - April 2003.

Park, S., X. Shen, J. Mark, and J. Pan, “Adaptive route optimization in hierarchical mobile IPv6 networks,” IEEE Transactions on Mobile Computing, accepted, 2006.

Rayside, D., M. Litoiu, M.-A. Storey, C. Best and R. Lintern, “Visualizing flow diagrams in websphere studio using SHriMP views (visualizing flow diagrams),” Information Systems Frontiers: A Journal of Research and Innovation, (Kluwer, ISSN 1387-3326), vol. 4, no. 4, 2002.

Rondogiannis, P., and W.W. Wadge, “Minimum model semantics for logic programs with negation-as-failure,” in ACM Transactions on Computational Logic, vol. 6, no. 2, Apr. 2005.

Ruskey, F., “A survey of Venn diagrams,” Electronic Journal of Combinatorics, DS #5, Second Edition, March 2001.

Ruskey, F., and J. Sawada, “Bent Hamilton cycles in d-dimensional grid graphs,” Electronic J. Combinatorics, vol. 10, no. R1, 18 pages, 2003.

Ruskey, F., and M. Weston, “More FUN with symmetric Venn diagrams,” Theory of Computing Systems, vol. 39, pp. 413-423, 2006.

Ruskey, F., C.D. Savage, and S. Wagon, “The search for simple symmetric Venn diagrams,” Notices of the American Mathematical Society,” pp. 1304-1311 (also the cover illustration), Dec. 2006.

Ruskey, F., C.R. Miers, and J. Sawada, “The number of irreducible polynomials and lyndon words with given trace,” SIAM J. Discrete Mathematics, 14, pp. 240-245, 2001.

Scholz, B., E. Mehofer, and R.N. Horspool, “Predicated partial redundancy elimination using a cost analysis,” Parallel Processing Letters, vol. 13, 4, pp. 525-536, 2003.

Sen, P., and S. Venkatesh, “Lower bounds in the quantum cell probe model,” Journal of Computer Systems and Sciences, to appear.

Storey M.-A., “Designing a software exploration tool using a cognitive framework of design elements,”in Software Visualization, March 2003.

Storey, M.-A., “Theories, tools and research methods in program comprehension: past, present and future,” Software Quality Journal, Springer, 2006.

Storey, M.-A., B. Phillips, M. Maczewski, and M. Wang, “Evaluating the usability of web-based learning tools,” in Educational Technology & Society, Journal of International Forum of Educational Technology & Society and IEEE Learning Technology Task Force, special issue on Evaluation of Learning Technologies, special guest editor, Grainne Conole, ISSN 1436-4522, Jul. 2002.

Storey, M.-A., K. Wong and H.A. Müller, “How do program understanding tools affect how programmers understand programs?” Science of Computer Programming, vol. 36 (2-3), pp. 183-207, Mar. 2000.

Sun, B., C.-C Li, K. Wu, and Y. Xiao, “A lightweight secure protocol for wireless sensor networks,” accepted by Elsevier Ad Hoc Networks Journal, Special issue on Security in Ad Hoc and Sensor Networks: Performance, Reliability, Security, and Beyond, 2005.

Sun, B., F. Yu, K. Wu, Y. Xiao, and V.C.M Leung, “Mobility-based anomaly detection in cellular mobile networks,” IEEE Transactions on Vehicular Technology, vol. 55, no. 3. May 2006.

Sun, B., K. Wu, U.W. Pooch, “Zone-based intrusion detection for mobile ad hoc networks,” accepted by Int. Journal on Ad Hoc & Sensor Wireless Networks, 2005.

Sun, B., K. Wu, Y. Xiao, and R. Wang, “Integration of mobility and intrusion detection for wireless Ad Hoc networks,” accepted by Wiley Int. Journal of Communication Systems, 2006.

Systä, T., K. Koskimies, and H.A. Müller, “Shimba - an environment for reverse engineering Java software systems,” Software - Practice and Experience, vol. 31, no. 4, pp. 371-394, Apr. 2001.

Thornton, M. A., and D. M. Miller, “Computation of Discrete Function Chrestenson Spectra Using Cayley Color Graphs,” Journal of Multiple-Valued Logic and Soft Computing, Special Issue on Spectral Techniques, vol. 10, no. 2, pp. 189-202, 2004.

Tilley, S.R., J. Gerdes, T. Hamilton, S. Huang, H.A. Müller, D. Smith, and K. Wong, “On the Business Value and Technical Challenges of Adopting Web Services,” Journal of Software Maintenance and Evolution: Research and Practice, 16(12):31-50, Jan./April 2004.

Tzanetakis, G., A. Ermolinskyi, and P. Cook, “Pitch histograms in sysmbolic and audio music information retrieval,” Journal of New Music Research, 32(2), pp. 143-152, 2003.

Tzanetakis, G., and P. Cook, “Music analysis and retrieval systems,” in Journal of American Society for Information Science and Technology, 55 (12), pp. 1077-1083, 2004.

Tzanetakis, G., J. Gao, and P. Steenkiste, “A scalable peer-to-peer system for music content and information retrieval,” in Computer Music Journal, (Special Issue on Music Information Retrieval), 28 (2), pp. 24-33. 2004.

van Rooij, I., A. Schactman, H. Kadlec, and U. Stege, “Perceptual or analytical processing? Evidence from children's and adult's performance on the Euclidean Traveling Salesman problem,” Journal of Problem Solving, vol. 1, no. 1, pp. 44-73, 2006.

van Rooij, I., U. Stege, and A. Schactman, “Convex hull and tour crossings in the Euclidean traveling salesperson problem: implications for human performance studies,” Memory and Cognition, vol. 31, no. 2, pp. 215-220, 2003.

van Rooij, I., U. Stege, and H. Kadlec, “Sources of complexity in subset choice,” Journal of Mathematical Psychology, vol. 49, no. 2, pp. 160-187, 2005.

Varadarajan, K., S. Venkatesh, J. Zhang, and Y. Ye, “On approximating the radii of joint sets in high dimensions,” SIAM Journal of Computing, to appear.

Wu, K., and J. Harms, “Performance study of proactive flow handoff for mobile ad hoc networks,” ACM Journal on Wireless Networks (WINE), vol. 12, no. 1, pp. 119-135, Feb. 2006.

Wu, K., D. Dreef, B. Sun, and Y. Xiao, “Secure data aggregation without persistent cryptographic operations in wireless sensor networks,” in Elsevier Ad Hoc Networks Journal, Special issue on Security in Ad Hoc and Sensor Networks, 2006.

Wu, K., Y. Gao, F. Li, and Y. Xiao, “Lightweight deployment-aware scheduling for wireless sensor networks,” ACM/Kluwer MONET Journal, Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks, vol. 10, no. 6, pp. 837-852, Dec. 2005.

Xiao, Y., H. Li, K. Wu, K. Leung, and Q. Ni, “On optimizing backoff counter reservation and classifying stations for the IEEE 802.11 distributed wireless LANs,” IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 17, no. 7, pp. 713-722, July 2006.

Xiao, Y., H. Li, Y. Pan, K. Wu, and J. Li, “On optimizing power consumption for mobile handsets,” IEEE Transactions on Vehicular Technology, vol. 53, no. 6, pp. 1927-1941, Nov. 2004.

Zastre, M., and R.N. Horspool., “Exploiting exceptions,” Software - Practice and Experience, vol. 31, 12, pp. 1109-1123, Oct. 2001.

B.2 Refereed Conference Publications 2000-2007

Agarwal, R., and W.W. Wadge, “Lazy evaluation of infinitesimal logic expressions,” 2005 Int. Conf. on Programming Languages and Compilers,” Las Vegas, Jun. 27-30, 2005.

Akanda, M.A.K., and D.M. German, “A component-oriented framework for the implementation of navigational design patterns,” in Proc. of the Int. Conf. on Web Engineering, pp. 449-450, July 2003.

Akanda, M.A.K., D.M. German, “A system of patterns for web navigation,” in Web Engineering: 5th Int. Conf. ICWE 2005, pp 136-141, July 2005.

Akbar, M.M., E.G. Manning, and G.C. Shoja, “Admission control and quality adaptation in the distributed multimedia server system (DMSS),” Proc. of Internet Multimedia Management Systems, pp. 246-257, 2001 SPIE Conf., Denver, U.S.A., Aug.22-23, 2001.

Akbar, M.M., E.G. Manning, and G.C. Shoja, “Heuristic solutions for the multiple-choice multi-dimension knapsack problem,” presented in the 2001 Int. Conf. on Computational Science (ICCS 2001), San Francisco, California, U.S.A., in Lecture Notes in Computer Science, LNCS 2074, Springer, pp. 659-668, May 28-30, 2001.

Akbar, M.M., E.G. Manning, G.C. Shoja, “Application of UM-D to optimal server selection for content routing,” Proc. ICCIT 02, Dhaka, Bangladesh, IEEE, pp. 60-63, ISBN 984-32-0328-3, Dec. 2002.

Akbar, M.M., E.G. Manning, G.C. Shoja, “Distributed utility model for distributed multimedia server system,” Pro. ICCIT 02, Dhaka, Bangladesh, pp. 315-320, ISBN 984-32-0450-6, Dec. 2002.

Akbar, M.M., E.G. Manning, R.K. Watson, G.C. Shoja, S. Khan, K.F. Li, “Optimal admission controllers for service level agreements in enterprise networks,” Proc. SCI 2002, 6th World Multiconf. on Systemics, Cybernetics and Informatics, Orlando, Fla., Jul. 14-18, 2002.

Alber, J., H. Fan, M. Fellows, H. Fernau, R. Niedermeier, F. Rosamond, and U. Stege, “Refined search tree techniques for the planar dominating set problem,” 26th Int. Symp. on Mathematical Foundations of Computer Science (MFCS 2001), Aug. 27 - 31, 2001, Marianske Lazne, Czech Republic, LNCS 2163, pp. 111-122, 2001.

Aycock, J., and R. N. Horspool, “Directly executable Earley Parsing,” Proc. of CC 2001, Compiler Construction Conference, Genoa Italy, Springer-Verlag, LNCS vol. 2304, pp. 229-243, Apr. 2001.

Aycock, J., and R. N. Horspool, “Faster generalized LR parsing,” in Proc. of CC '99 Compiler Construction Conf., Amsterdam, Netherlands, Springer-Verlag, LNCS vol. 1575, pp. 32-46, Mar.1999.

Aycock, J., and R.N. Horspool, “Simple generation of static single assignment form,” in Proc. of CC2000, Compiler Construction Conference, Berlin, Germany, Springer-Verlag, LNCS vol. 1781, pp. 110-124, Mar./Apr. 2000

Aycock, J., and Zastre, M., “An exceptional programming language,” in Proc. 2005 Int. Conf. on Programming Languages and Compilers, H.R. (ed.) ISBN 1-932415-75-0, pp. 137-141, July 2005.

Baldwin, J., and Y. Coady, “Adaptive systems require adaptive support - when tools attack!”, in Proc. of the Hawaii Int. Conf. on System Sciences (HICSS), Jan 2007.

Baldwin, J., J. Zigman, and Y. Coady, “Version 2.*.* and counting! The toll of evolution on Aspect-oriented distribution,” in Proc. of LATE Workshop, held at the Int. Conf. on Aspect-Oriented Software Development (AOSD), Mar. 2006.

Barclay, B.M. Price, and J.H. Weber-Jahnke, “Open technology assisted practice application suite,” Int. Conf. on Information Technology and Communication in Health, Victoria, B.C., 2007.

Barksy, M., U. Stege, A. Thomo, and C. Upton, “A new algorithm for fast all against all substring matching,” Proc. of the 13th Symp. on String Processing and Information Retrieval (SPIRE' 06), Glasgow, Scotland, LNCS 4209, pp. 360-366, Oct. 11-13, 2006.

Berry, D.M., D. Damian, A. Finkelstein, D. Gause, R. Hall, E. Simmons, and A. Wassyng, “To do or not to do: if the requirements engineering payoff is so good, why aren't more companies doing it?” Proc. of the Int. Conf. on Requirements Engineering (RE06), Paris, p. 447, Aug. 2005.

Best, C., M.-A. Storey and J. Michaud, “Designing a Component-Based Framework for Visualization in Software Engineering and Knowledge Engineering”, Fourteenth International Conference on Software Engineering and Knowledge Engineering, Ischia, Italy, pp. 323-326, Jul. 15-19, 2002.

Bilykh, I., J.H. Jahnke, G. McCallum, M. Price, “Using the clinical document architecture as open data exchange format for interfacing EMRs with clinical decision support systems,” IEEE Conf. on Computer-Based Medical Systems (CBMS), Utah, USA, Jun. 2006.

Bilykh, I., Y. Bychkov, D. Dahle, J.H. Jahnke, G. McCallum, C. Obry, A. Onabajo, and C. Kuziemasky, “Can GRID Services provide answers to the challenges of National Health Record Sharing?” Proc. of the 2003 Conf. of the Centre for Advanced Studies on Collaborative Research (CASCON), IBM Center for Advanced Studies, Toronto, Canada, pp. 39-53, 2003.

Bingham, J., and M. Serra, “Solving Hamiltonian Cycle on FPGA Technology via Instance to Circuit Mappings,” Int. Workshop on Engineering of Reconfigurable Hardware/Software Objects (ENREGLE), Las Vegas, Jun. 2000.

Bingham,J., and J.C. Muzio, “Advanced evolution of nonlinear cellular automata for combinational BIST,” European Test Workshop, Stockholm, May 2001.

Bishop, J., and R.N. Horspool, “Towards principles of GUI programming,” Proc. of SIGCSE 2004, Norfolk, VA, ACM Press, pp. 373-377, Mar. 2004.

Bo, S., K. Wu, and U.W. Pooch, “Secure routing against black-hole attack in mobile ad hoc networks,” Proc. of LASTED Int. Conf. on Communications and Computer Networks (CCN 02), Boston, pp. 170-176, Nov. 2002.

Bonneville, E., J.C. Muzio, and M. Serra, “Usability Issues in Software to Assist People with Brain Injuries,” 6th European Consortium for Information and Mathematics (ERCIM), pp. 42-56, Oct. 2000.

Bray, S., and G. Tzanetakis, “Distributed audio feature extraction for music,” in Proc. of the Int. Conf. on Music Information Retrieval (ISMIR), 2005.

Bray, S., and G. Tzanetakis, “Implicit patching for dataflow-based audio analysis and synthesis,” in Proc. of the Int. Computer Music Conf. (ICMC), 2005.

Brodsky, A., D. Brodsky, I. Chan, Y. Coady, S. Gudmundson, J. Pomkoski, and J.S. Ong, “Coping with evolution: aspects vs aspirin?” in Proc. of Workshop on Advanced Separation of Concerns, held at the ACM Conf. on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Oct. 2001.

Brown, R.H.F , and R.N. Horspool, “Object-specific redundancy elimination techniques,” Proc. of Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS's006), July 3, 2006.

Bryce, R.W., G.C. Shoja, “Chameleon: a new operating system concept for supporting multimedia,” Proc. of Int. Conf. on Internet and Multimedia Systems and Applications (IMSA 2003), Honolulu, Hawaii, pp. 20-25, Aug. 13-15, 2003.

Bull, I., M.-A. Storey, M. Litoiu, and J.M. Favre, “An architecture to support model driven software visualization,” 14th IEEE Int. Conf. on Program Comprehension (ICPC 2006), (short paper), Athens, Greece, Jun. 14-16, 2006.

Bull, R.I., C.A. Best, and M.-.A. Storey, “Advanced widgets for eclipse,” Eclipse Technology Exchange and Workshop, 19th Annual ACM Conf. on Object-Oriented Programming, Systems, Languages, and Applications, Vancouver, Canada , Oct. 24-28, 2004.

Burroughs, N., A. Parkin, G. Tzanetakis, “Flexible scheduling for dataflow audio processing,” in Proc. of the Int. Computer Music Conf. (ICMC), 2006.

Buss, S., and B.M. Kapron, “Resource-bounded continuity and sequentiality for type-two functionals,” in Proc. 15th IEEE Symp. on Logic in Computer Science, pp. 77-83, 2000.

Bychkov, Y., and J.H. Jahnke, “Interactive migration of legacy databases to net-centric technologies,” in Proc. of Working Conf. on Reverse Engineering (WCRE 2001), IEEE-CS Press, Stuttgart, Germany, pp. 328-334, 2001.

Bychkov, Y., and J.H. Jahnke, “VARLEL/Babel: toolkit for net-centric legacy data integration,” in Proc of 3rd Workshop on Net-Centric Computing (NCC'01), (co-located with ICSE 01), 8 pages, Toronto, Ontario, 2001.

Byres, D., B. Chauvin, D. Hoffman, J. Karsch, and N. Kube, “The special needs of SCADA/PCN firewalls: architectures and test results,” in Proc. of Int. Conf. on Emerging Technologies and Factory Automation, Catania, Italy, Sep. 2005.

Byres, E., D. Hoffman, and N. Kube, “On shaky ground: a study of security vulnerabilities in control protocols,” in Proc. of 5th American Nuclear Society Int. Topical Meeting on Nuclear Plant Instrumentation, Controls and Human Interface Technology, Albuquerque, NM, 2006.

Byres, E., J. Carter, A. El Ramly, and D. Hoffman, “Worlds in collision - Ethernet and the factory floor,” in Proc. ISA Emerging Technologies Conf., ISA Vol. 25, Oct. 2002.

Cai, L., J. Pan, and X. Shen, “Vulnerability analysis of IP traceback schemes,” in Proc. of 48th IEEE Globe Telecommunications Conf. (GLOBECOM05), St. Louis, MO, USA, pp. 1806-1810, Nov. 28-Dec. 2, 2005.

Cai, L., J. Pan, X. Shen, and J. Mark, “Peer collaboration in wireless ad hoc networks,” in Proc. of 4th IFIP Int. Conf. on Networking (Networking05), Waterloo, Ontario, Canada, pp. 840-852, May 2-6, 2005.

Cai, L., X. Shen, J. Mark, and J. Pan, “QoS-Aware AIMD protocols for multimedia services in wireless/IP networks,” in Proc. of 24th IEEE Int. Conf. on Computer Communications (INFOCOM05), Miami, FL, USA, pp. 2008-2019, Mar. 15-17, 2005.

Cai, M., J. Pan, K. Kwok, and K. Hwang, “Fast and accurate traffic matrix measurement using adaptive cardinality counting,” in Proc. of 2005 ACM SIGCOMM Workshop on Mining Network Data (MineNet05), Philadelphia, PA, USA, pp. 205-206, Aug. 26, 2005.

Calabar, P., D., Pearce, P. Rondogiannis, W. Wadge, “A purely model-theoretic Semantics for disjunctive logic programming with negation,” accepted to LPNMR07, Tempe, Arizona, May 2007.

Carruthers, S., and V. King, “Connectivity of wireless sensor networks with constant density,” Proc. of 3rd Int. Conf. ADHOC-NOW, Springer Lecture Notes in Computer Science, vol. 3158, pp. 149-157, Aug. 2004.

Cheng, B., R. de Lemos, S. Fickas, D. Garlan, J. Magee, M. Litoiu, H.A. Müller, and R. Taylor, “SEAMS 2007: Software Engineering for Adaptive and Self-Managing Systems,” Proceedings 29th IEEE/ACM Int. Conf. on Software Engineering (ICSE 2007), Minneapolis, Minnesota, USA; May 26-27, 2007, 2 pages, May 2007.

Cheng, M., P. Miskiewicz, K. Wu, and M. Zastre, “Fresh as a ROSE: Java + C in an introductory OS course,” Western Canadian Conf. on Computing Education, Courtenay, BC, May 2003.

Chisan, J., and D. Damian, “Towards a model of awareness support of software development in GSD,” Proc. of 3rd Int. Workshop on Global Software Development, ICSE'04, Scotland, 2004.

Chow, S., and F. Ruskey, “Area-proportional drawings of intersecting families of simple closed curves,” Proc. of 17th Canadian Conf. on Computational Geometry, pages 118-121, 2005.

Chow, S., and F. Ruskey, “Drawing area proportional Venn and Euler diagrams,” 11th Int. Symp. on Graph Drawing, Perugia, Italy, 2003, Lecture Notes in Computer Science, no. 2912, pp. 466-477, 2004.

Chow, S., and F. Ruskey, “Towards a general solution to drawing area-proportional Euler diagrams, Electronic Notes in Theoretical Computer Science, Proc. of 1st Int. Workshop on Euler Diagrams (Euler 2004), edited by P. Rodgers, Sept. 22-23, 2004.

Coady, Y., A. Brodsky, D. Brodsky, J. Pomkoski, S. Gudmundson, J.S. Ong, and G. Kiczales, "Can AOP support extensibility in client-server architectures?" in Proc. of Aspect-Oriented Programming Workshop, held at the European Conf. on Object-Oriented Programming (ECOOP), Jun. 2001.

Coady, Y., and C. Gibbs, "Hey! You've got your Aspects in my undergraduate curriculum," in Proc. of AIT Workshop, held at the Int. Conf. on Aspect-Oriented Software Development (AOSD), Mar. 2006.

Coady, Y., G. Kiczales, J.S. Ong, A. Warfield, and M. Feeley, "Brittle systems will break – not bend: can AOP help?" in Proc. of 10th ACM SIGOPS European Workshop on Operating Systems, Sep. 2002.

Coady, Y., G. Kiczales, M. Feeley, and G. Smolyn, "Using AspectC to improve the modularity of path-specific customization in operating system code," in Proc. of Joint European Software Engineering Conf. (ESEC), and 9th ACM SIGSOFT Int. Symp. on the Foundations of Software Engineering (FSE-8), pp. 88-98, Sep. 2001.

Coady, Y., G. Kiczales, M. Feeley, N. Hutchinson, J.S. Ong, and S. Gudmundson, "Exploring an aspect-oriented approach to OS code," in Proc. of 4th ECOOP Workshop on Object-Orientation and Operating systems, held at the European Conf. on Object-Oriented Programming (ECOOP), Jun. 2001.

Coady, Y., G. Kiczales, M. Feeley, N. Hutchinson, J.S. Ong, and S. Gudmundson, "Position summary: aspect-oriented system structure," in Proc. of 8th Workshop on Hot Topics in Operating Systems (HotOS), May 2001.

Coady, Y., G. Kiczales, M. Feeley, N. Hutchinson, J.S. Ong, and S. Gudmundson, "Structuring system aspects," in Proc. of Advanced Separation of Concerns Workshop, held at the Int. Conf. on Software Engineering (ICSE), May 2001.

Corrie, B. and M.-A. Storey, "Towards understanding the importance of gesture in distributed collaborative environments," ICMI05 Workshop on Multimodal Interaction for the Visualization and Exploration of Scientific Data, Trento, Italy, Oct. 3, 2005.

Costi, C., and D.M. Miller, "A Visualization Framework for VHDL Analysis," Fifth Multi-Conference on Systems, Cybernetics and Informatics: Special Session on Modern Digital System Synthesis, pp. 278-282, Orlando, Fl., July 2001.

Crider, M., S. Bergner, T. Smyth, T. Möller, M. Tory, A. Kirkpatrick, and D. Weiskopf, "A mixing board interface for graphics and visualization software," Graphics Interface 2007, accepted, Feb. 2007.

Cubranic, D., and M.-A. Storey, "Collaboration support for novice team programming," GROUP 2005, Florida, USA, pp. 136-139, Nov. 2005.

Cubranic, D., M.-A. Storey, and J. Ryall, "A comparison of communication technologies to support novice team programming," ICSE education track (short paper), Shanghai, China, May 2006.

Dahlem, D., J.H Jahnke, A. Onabajo, and Q.L. Wang, "The challenges of implementing web services on small devices," in Proc. of OOPSLA 2002 Workshop on Pervasive Computing, Seattle, 8 pages, Nov. 2002.

Dahlem, D., Y. Bychkov, L. Kawasme, and J.H. Jahnke, "Towards context aware web services for smart personal object technologies," in Proc. of OOPSLA Workshop on Pervasive Computing, 7 pages, 2003.

Damian D., "The use of a multimedia group support system for distributed software requirements meetings," in Proc. of 35th Hawaii Conf. on Systems Sciences, Hawaii, Jan. 2002.

Damian Herlea, D.E., A. Eberlein, M.L.G. Shaw, and B.R. Gaines, "A study of requirements negotiations in virtual project teams," in Proc. of 8th European Conf. on Information Systems, Vienna, Austria, pp. 937-944, 2000.

Damian Herlea, D.E., A. Eberlein, M.L.G. Shaw, and B.R. Gaines, "The effects of communication media on group performance in requirements engineering," extended abstract, Proc. of 4th Int. Conf. on Requirements Engineering, Schaumburg, Illinois, USA, p. 191, 2000.

Damian Herlea, D.E., M.L.G. Shaw, B.R. Gaines, and D. Zowghi, "A multi-disciplinary approach to the study of distributed requirements negotiations," Proc. of 5th Australian Workshop on Requirements Engineering, Brisbane, Australia, Dec. 8-9, 2000.

Damian, D., "A research methodology in the study of requirements negotiations in geographically distributed software teams," Proc. of the Workshop on Comparative Evaluation in RE (CERE), Int. Conf. on Requirements Engineering, Monterey Bay, Sep. 8-12, 2003.

Damian, D., "An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective?" Proc. of Asia Pacific Software Engineering Conf. (APSEC), Macau, Dec. 2001.

Damian, D., "Negotiation behavior and group interaction in face-to-face and distributed requirements negotiations: four case studies," Proc. of Australian Workshop on Requirements Engineering, Sydney, Australia, Nov. 2001.

Damian, D., "The study of requirements engineering in global software development: as challenging as important," Proc. of Int. Workshop on Global Software Development, Int. Conf. on Software Engineering (ICSE), Florida, USA, 2002.

Damian, D., A. Eberlein, B. Woodward, M. L.G. Shaw and B.R. Gaines, "An empirical study of facilitation of computer-mediated distributed requirements negotiations," Proc. of Requirements Engineering Symp. (RE 01), Toronto, Canada, Aug. 2001.

Damian, D., A. Hadwin, and B. Al-Ani, "Instructional design and assessment strategies for teaching global software development: a framework," Proc. of Int. Conf. on Software Engineering, Shanghai, May 20-28, 2006.

Damian, D., and A. Eberlein, "Groupware support for requirements negotiations in distributed software development," Proc. of 4th ICSE Workshop on Software Engineering over the Internet, Toronto, Canada, May 2001.

Damian, D., and D. Zowghi, "An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations," Proc. of the 36th Hawaii Int. Conf. on Systems Sciences, Collaboration Systems and Technology Track, Hawaii, Computer Society, 2003.

Damian, D., and D. Zowghi, "The impact of stakeholders' geographical distribution on requirements engineering in a multi-site development organization," Proc. of the 10th IEEE Int. Conf. on Requirements Engineering (RE'02), Essen, Germany, pp. 319-328, 2002.

Damian, D., and J. Chisan, "Exploring the role of requirements engineering in improving risk management," Proc. of Int. Conf. on Requirements Engineering, RE'06, pp. 481-482, Paris, Aug. 2005.

Damian, D., B. Al-Ani, D. Cubranic, and L. Robles, "Teaching requirements engineering in global software development: a report on a three-university collaboration," Proc. of Workshop on Requirements Engineering Education and Training, Paris, Aug. 2005.

Damian, D., D. Zowghi, L.Vaidyanathasamy, and Y. Pal, "An industrial experience in process improvement: an early assessment at the Australian Centre for Unisys Software," Proc. of the Int. Symp. on Empirical Software Engineering, Nara, Japan, pp. 111-126, 2002.

Damian, D., F. Lanubile, and D. Mallardo, "The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations," Proc. of Int. Conf. on Software Engineering, Shanghai, May, 685-690, 2006.

Damian, D., F. Lanubile, and T. Mallardo, "An empirical study of the impact of asynchronous discussions on remote synchronous requirements meetings," Proc. of Fundamental Approaches to Software Engineering (FASE), Vienna, March 27-29, 2006.

Damian, D., F. Lanubile, and T. Mallardo, "Investigating IBIS in a distributed educational environment: the design of a case study," Proc. of Int. Workshop on Distributed Software Engineering, Paris, Aug. 2005.

Damian, D., J. Chisan, L. Vaidyanathasamy, Y. Pal, "An industrial case study of the impact of requirements engineering on downstream development," Proc. of the Int. Symp. on Empirical Software Engineering (ISESE), Rome, Italy, 2003.

Damian, D., J. Chisan, P. Allen, and B. Corrie, "Awareness meets requirements management: awareness needs in global software development," Proc. Int. Workshop on Global Software Development, Int. Conf. on Software Engineering, Portland, USA, 2003.

Damian, D.E., A. Eberlein, B. Woodward, M.L.G. Shaw, and B.R. Gaines, "An empirical study of facilitation of computer-mediated distributed requirements negotiations," Proc. of 5th Int. Symp. on Requirements Engineering, Toronto, Canada, Aug. 2001.

Dannenberg, R., W. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo, “The MUSART testbed for query-by-humming evaluation,” in Proc. of the Int. Conf. on Music Information Retrieval (ISMIR) Baltimore US, 2003.

De Souza, P., A. Mcnair, and J.H. Jahnke, “Network-centric migration of embedded control software - a case study,” Proc. of the 2003 Conf. of the Centre for Advanced Studies on Collaborative Research, (CASCON), IBM Center for Advanced Studies, Toronto, Canada, pp. 54-65, 2003.

d'Entremont, M., and J.H. Jahnke, “Dynamic embedded network configuration,” in Proc. of OOPSLA 2002 Workshop on Pervasive Computing, Seattle, Nov. 2002.

d'Entremont, M., and J.H. Jahnke, “MicroSynergy - generative tool support for networking embedded controllers,” in Proc. of 3rd Workshop on Net-Centric Computing (NCC'01), (co-located with ICSE 01), 8 pages, Toronto, Ontario, 2001.

Desmarais, R., and H.A. Müller, “A Proposal for an Autonomic Grid Management System,” Proc. Intl. Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2007); Workshop at 29th IEEE/ACM Int. Conf. on Software Engineering (ICSE 2007), Minneapolis, Minnesota, USA; May 26-27, 2007, 7 pages, May 2007.

Desmond, M., M.-A. Storey, and C. Exton, “Fluid source code views for Just In-Time Comprehension,” SPLAT 2006: Software Engineering Properties of Languages and Aspect Technologies, a workshop affiliated with AOSD 2006, Bonn, Germany, Mar. 21, 2006.

Deugau, C. and F. Ruskey, “Complete k-ary trees and generalized meta-Fibonacci sequences,” 4th Colloquium on Mathematics and Computer Science: Algorithms, Trees, Combinatorics and Probabilities, Sept. 18-22, 2006, Institut Élie Cartan, Nancy, France, DMTCS Proc. Series, vol. AG, pp. 203-214, 2006.

Dreef, D., S. Ahari, K. Wu, and V. King, “Utilizing the uncertainty of intrusion detection to strengthen security for ad-hoc networks,” Proc. of 3rd Int. Conf. on AD-HOC Networks & Wireless (ADHOC-NOW 04), LNCS3158, Vancouver, pp. 82-95, May 2004.

Du, Y., and D. Hoffman, “Pbit - a pattern-based testing framework for iptables,” in Proc. of Communication Networks and Services Conf., May 2004.

Dubrova, E., P. Ellervee, D.M. Miller, and J.C. Muzio, “TOP: an algorithm for three-level optimization of PLDs,” Proc. Design, Automation and Test in Europe Conf., pp. 751, Mar. 2000.

Dubrova, E., P. Everlee, D.M. Miller and J.C. Muzio, “TOP, An Algorithm for Three-Level Minimization of PLDs,” Proc. Design, Automation and Test in Europe Conference, pp. 751, March 2000.

Dueck, G.W., D. Maslov, and D.M. Miller, “A Transformation-based Algorithm for Reversible Logic Synthesis,” IEEE/ACM Design Automation Conference (DAC), Anaheim, CA, June 2-6, 2003, pp. 318-323.

Dueck, G.W., D. Maslov, and D.M. Miller, “Transformation-based Synthesis of Networks of Toffoli/Fredkin Gates,” IEEE Canadian Conference on Electrical and Computer Engineering, Montreal, Quebec, May 2003.

Dulac, N., T. Viguier, N. G. Leveson, and M.-A. Storey, “On the use of visualization in formal requirements specification,” Proceedings of the IEEE Joint International Requirements Engineering Conference, University of Essen, Germany, pp. 71-80, Sep. 9-13, 2002.

Eden, A., and J.H. Jahnke, “Coordinating software evolution via two-tier programming,” in Proc. of 5th Int. Conf. on Coordination Models and Languages, vol. 2315 of Lecture Notes in Computer Science, pp. 149-159, Springer Verlag,York, UK, Apr. 8-11, 2002.

El-Kharashi, M.W., G. Darling, B. Marykuca, and G.C. Shoja, “Data link layer: a lab project in computer communications and networks for fourth year undergraduate students in computer science and engineering,” Proc. of the 12th Canadian Conf. on Engineering Education, Victoria, B.C., pp. 282-288, Aug. 23-25, 2001.

Farfeleder, S., A. Krall, and N. Horspool, “Ultra fast cycle-accurate compiled emulation of inorder pipelined architectures,” Proc. of SAMOS V: Embedded Computer Systems: Architectures, MOdeling, and Simulation, Samos, Greece, LNCS vol. 3553, Springer-Verlag, pp. 222-231, July 2005.

Feldstein, C.B. and J.C. Muzio, “Development of a fault tolerant flight control system,” Proc. of 23rd Digital Avionics Systems Conf., (best paper award), 2004.

Fellows, M., C. Knauer, N. Nishimura, P. Ragde, F. Rosamond, U. Stege, D. Thilikos, and S. Whitesides, “Faster fixed-parameter tractable algorithms for matching and packing problems,” 12th Annual European Symp. on Algorithms (ESA'04), Bergen, Norway, LNCS 3221, pp. 311-322, 2004.

Fellows, M.R., C. McCartin, F. Rosamond, and U. Stege, “Coordinatized kernels and catalytic reductions: an improved FPT algorithm for max leaf spanning tree and other problems, Foundations of Software Technology and Theoretical Computer Science (FST TCS 2000), pp. 240-251, Dec. 2000.

Fellows, M.R., C. McCartin, F. Rosamond, and U.Stege, “Coordinatized kernels and catalytic reductions: an improved FPT algorithm for max leaf spanning tree and other problems,” Eleventh Australian Workshop on Combinatorial Algorithms (AWOCA 2000), Jul. 2000.

Fowler, P.W., and W. Myrvold, “Some chemical applications of independent sets,” in Proc. of 18th Mini Euro Conf. on Variable Neighborhood Search, P. Hansen, N. Mladenovic, J.A.M. Perez, B.M. Batista, and J.M. Moreno-Vega (editors), Tenerife, Spain, 12 pages, 2005.

Gagarin, A., W. Myrvold, and J. Chambers, “Forbidden minors and subdivisions for toroidal graphs with no K 3,3's, Electronic Notes, Discrete Mathematics, vol. 22, pp. 151-156, 2005.

Gaines, B., and D. Damian, “Framing the requirements for research environments of the future,” BCNet Advanced Networks Conf. 2004, Theme: e-Science, Track: Projects of the Future, Vancouver, 2004.

Galanaki, P., P. Rondogiannis, and W.W. Wadge, “Game semantics for infinite propositional logic programs with negation,” (GaLoP'06).

Ganti, S., and B. Raheemi, “Differentiated back-off for ethernet,”IEEE Canadian Conf. and Electrical and Computer Engineering, Saskatoon, May 2005.

Gao, J., G. Tzanetakis, and P. Steenkiste, “Content-based retrieval of music in scalable peer-to-peer networks,” in Proc. of the Int. Conf. on Multimedia and Expo (ICME), Baltimore, US, 2003.

Gao, Y., K. Wu, and F. Li, “Analysis on the redundancy of wireless sensor networks,” Proc. of 2nd ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA 03), San Diego, California, pp. 108-114, Sep. 2003.

German, D., “The evolution of the GNOME Project,” in Proc. of the 2nd Workshop on Open Source Software Engineering, May 2002.

German, D., and D. Cowan, “Towards a unified catalog of hypermedia design patterns,” in Proc. of the 33rd Hawaii Int. Conf. on System Sciences, Hawaii, U.S.A., Jan. 2000.

German, D., P. Rigby, and M.-A. Storey, “Using evolutionary annotations from Change Logs to Enhance Program Comprehension,” MSR 2006: International Workshop on Mining Software Repositories (co-located with ICSE 2006), Shanghai, China, May 22-23, 2006.

German, D.M., “A study of the contributors of PostgreSQL,” in 3rd Int. Workshop on Mining Software Repositories - MSR Challenge Reports (MSR 2006), received Best Challenge Report Award, May 2006.

German, D.M., “An empirical study of fine-grained software modifications,” in 20th IEEE Int. Conf. on Software Maintenance (ICSM'04), Sept. 2004.

German, D.M., “Experiences teaching a graduate course in open source software engineering,” in Open Educational Symp. of 1st Int. Conf. on Open Source Systems, July 326-329, 2005.

German, D.M., “GNOME, a case of open source global software development,” in Proc. of the Int. Workshop on Global Software Development, May 2003.

German, D.M., “Mining CVS repositories, the softChange experience,” in 1st Int. Workshop on Mining Software Repositories, pp. 17-21, 2004.

German, D.M., “The flow of knowledge in free and open source communities,” 2nd Int. Workshop in Supporting Knowledge Collaboration in Software Development (KCSD 2006), to be presented, Sept. 2006.

German, D.M., “Using HadeZ to formally specify the Web museum of the National Gallery of Art,” in 2nd Int. Workshop on Web Oriented Software Technology, June 2002.

German, D.M., A. Hindle, and N. Jordan, “ Visualizing the evolution of software using softChange,” in Proc. SEKE 2004, 16th Int. Conf. on Software Engineering and Knowledge Engineering, Knowledge Systems Institute, 3420 Main St., Skokie IL 60076, USA, pp. 336-341, June 2004.

German, D.M., and A. Hindle, “Measuring fine-grained change: towards modification aware change metrics,” in Metrics 2005: 11th IEEE Int. Metrics Symp., Sept. 2005.

German, D.M., and A. Mockus, “Automating the measurement of open source projects,” in Proc. of the 3rd Workshop on Open Source Software Engineering, May 2003.

German, D.M., and P. Rigby, and M.-A. Storey, “Using evolutionary annotations from change logs to enhance program comprehension,” 3rd Int. Workshop on Mining Software Repositories (MSR 2006), May 2006.

German, D.M., D. Cubranic, and M.-A. Storey, “A framework for describing and understanding mining tools in software development,” in 2nd Int. Workshop on Mining Software Repositories, pp. 95-99, May 2005.

German, D.M., J. Robles, J.M. Gonzalez-Baharona, “The challenges of automated quantitative analysis of open source software projects,” Workshop on Evaluation Frameworks for Open Source Software (EFOSS), Apr. 2006.

German, D.M., Robles G., Gonzales-Baharona J.M., “The challenges of quantitative analysis of open source software projects, EFOSS Workshop at the 2nd Open Source Conf., June 2006.

Ghobadi, M., S. Ganti, S., G.C. Shoja, “Resource optimization to provision a virtual private network using the hose model,” IEEE ICC 2007, accepted for publication, Glasgow, June 2007.

Gibbs, C., and Y. Coady, “Dynamic patient care with PHACTS,” in Proc. of the Int. Conf. addressing Information Technology and Communications in Health (ITCH), Feb. 2007, (to appear).

Gibbs, C., and Y. Coady, “Garbage collection in jikes: could dynamic aspects add value?” in Proc. of 1st AOSD Aspects Workshop (DAW'04), held at the Int. Conf. on Aspect-Oriented Software Development, Mar. 2004.

Gibbs, C., and Y. Coady, “Making real-time abstractions concrete with aspects,” 3rd Workshop on Java Technologies for Real-time and Embedded Systems, held at the ACM Conf. on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), Oct. 2005.

Gibbs, C., and Y. Coady, “OASIS: organic aspects for system infrastructure software easing evolution and adaptation through natural decomposition,” in Proc. of the Hawaii Int. Conf. on System Sciences (HICSS), Jan. 2005.

Gibbs, C., and Y. Coady, “OASIS: organic aspects for system infrastructure software,” Workshop on Reflection, AOP and Meta-Data for Software Evolution, (held at the European Conf. on Object-Oriented Programming (ECOOP), June 2004.

Gibbs, C., and Y. Coady, “SAUSI: System aspect for uniformity in software infrastructure,” in Proc. of the 17th Int. Conf. on Software and Systems Engineering and their Applications (ICSSEA), Dec. 2004.

Gibbs, C., D. Lohmann, and Y. Coady, “Modular integration through aspects: making cents of legacy systems,” in Proc. of the Hawaii Int. Conf. on System Sciences (HICSS), Jan. 2007.

Gibbs, C., P. Proctor, and Y. Coady, “Surrendering to the need for speed while maintaining visibility in adverse code conditions,” in Proc. of LATE Workshop, held at the Int. Conf. on Aspect-Oriented Software Development (AOSD), Mar. 2007 (to appear).

Gibbs, C., R. Liu, and Y. Coady, “And the band played on: are aspects adrift in a sea of sinking code?” Workshop on Linking Aspect Technology and Evolution, held at the Int. Conf. on Aspect-Oriented Software Development (AOSD), Mar. 2005.

Gibbs, C., Y. Coady, J. Vitek, T. Zhao, C. Andreae, and J. Noble, “It is time to get real with Real-time: in search of the balance between tools, patterns and aspects,” in Proc. of ACP4IS Workshop, held at the Int. Conf. on Aspect-Oriented Software Development (AOSD), Mar. 2006.

Gibbs, C., Y. Coady, M. Haupt, J. Vitek and H. Yamauchi, “Towards a domain specific aspect language for virtual machines,” in Domain Specific Aspect Languages (DSAL) Workshop, held at the ACM Conf. on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Oct. 2006.

Grahne, G., A. Thomo, and W. Wadge, “Preferentially annotated regular path queries,” Proc. of the 11th Int. Conf. on Database Theory (ICDT07), Springer, 2007.

Guersesa, S., J.H. Jahnke, C. Obry, A. Onabajo, T. Santen, and M. Price, “Eliciting confidentiality requirements in practice,” Conf. of the Centre of Advanced Studies (CASCON), IBM Press, ACM Digital Library, 2005.

Gui, G., H.M. Kienle, and H.A. Müller, "REGoLive: Web Site Comprehension with Viewpoints," 13th IEEE International Workshop on Program Comprehension (IWPC 2005), St. Louis, MO; pp. 161-164, May 15-16, 2005.

Gui, G., H.M. Kienle, and H.A. Müller, "ReGoLive: Building a Web Site Comprehension Tool by Extending GoLive," IEEE International Conference on Web Site Evolution (WSE 2005), Budapest, Hungary, pp. 46-56, Sep. 26, 2005.

Hargreaves, E., and D. Damian, "Can global software teams learn from military teamwork models?" Proc. of 3rd Int. Workshop on Global Software Development, ICSE'04, Scotland, 2004.

Harsha, P., Y. Ishai, J. Kilian, K. Nissim, and S. Venkatesh, "Communication vs computation," Proc. of 31st Int. Colloquium of Automata Languages and Programming (ICALP), pp. 745-756, 2004.

Hauptman, Al., et al., "Informedia at TRECVID 2003: analyzing and searching broadcast news video," in Proc. of the (VIDEO) TREC 2003, 12th Text Retrieval Conf., Nov. 2003.

Hechmer, A., and G. Tzanetakis, "LogoRhythms: introductory audio programming for computer musicians," in Proc. of the Frontiers in Education Conf., 2006.

Higham, L., and L. Jackson, " Translating between Itanium and space memory consistency models," Symp. on Parallelism in algorithms and architectures, Jul.-Aug. 2006.

Higham, L., L. Jackson, and J. Kawash, "Programmer-centric conditions for Itanium memory consistency," Intl. Conf. on Distributed Computing and Networking.

Higham, L., L. Jackson, and J. Kawash, "What is Itanium memory consistency from the programmers point of view?" TV06: Multithreading in Hardware and Software: Formal Approaches to Design and Verification," Aug. 2006.

Hindle, A., and D.M. German, "A formal model and a query language for source control repositories," in 2nd Int. Workshop on Mining Software Repositories, pp. 100-105, May 2005.

Hoffman, D. and N. Kube, Automated testing of SCADA protocols, in Proc. of SCADA security Scientific Symp., Miami, Florida, 2007.

Hoffman, D., "Software engineering education: difficult decisions in turbulent times," in Proc. Canadian Conf. on Engineering Education, Aug. 2001.

Hoffman, D., and K. Yoo, "Blowtorch: a framework for firewall test automation," in Proc. of Int. Conf. on Automated Software Engineering, California, Nov. 2005.

Hoffman, D., and P. Strooper, "Prose + test cases = specifications," in Proc. 34th Int. Conf. on Technology of Object-Oriented Languages and System, pp. 239-250, IEEE Computer Society, Aug. 2000.

Hoffman, D., and P. Strooper, "Tools and techniques for Java API testing," in Proc. Australian Software Engineering Conf., IEEE Computer Society, July 2000.

Hoffman, D., and S. Wilkin, "JUnit extensions for documentation and inheritance," in Proc. Pacific Northwest Software Quality Conf., pp. 72 - 84, Oct. 2002.

Hoffman, D., D. Prabhakar, and P. Strooper, "Testing ip tables," in Proc. of CASCON 2003, Oct. 2003.

Hoke, Y., and W.W. Wadge, "An authoring tool for temporal intensional web pages," Int. Conf. on Programming Languages and Compilers," Las Vegas, Jun. 27-30, 2005.

Holtby, D., B. Kapron, and V. King, "Lower bounds for scalable Byzantine agreement," Proc. of the 23rd Annual ACM Principles of Distributed Computing (PODC), ACM, pp. 285-291, July 2006.

Holtby, D., B.M. Kapron, and V. King, "Lower bounds for scalable Byzantine agreement," in Proc. 25th ACM Symp. on Principles of Distributed Computing, pp. 285-291, 2006.

Hornsberger, J., an G.C. Shoja, "Geographic grid routing for wireless sensor networks," in Proc. of 2005 IEEE Int. Conf. on Networking, Sensing and Control, Tucson, Arizona, USA, pp. 484-489, Mar. 19-22, 2005.

Horspool, R.N., and J. Aycock, "Analysis of equation structure using least-cost parsing," in Proc. of IWPT 2000, 6th Int. Workshop on Parsing Technology, Trento, Italy, pp. 307-308, Feb. 2000.

Horspool, R.N., D.J. Pereira, and B. Scholz, "Fast profile-based partial redundancy elimination,: Proc. of Joint Modular Languages Conf,. JMLC'06, Springer-Verlag LNCS 4228, pp. 362-376, Sept. 13-15, 2006.

Hou, Y., J. Pan, and X. Shen, “On randomized request redirection in hierarchical caching systems,” in Proc. of 4th IEEE Globe Telecommunications Conf. (GLOBECOM03), San Francisco, CA, USA, pp. 3835-3839, Dec. 1-5, 2003.

Hou, Y., J. Pan, B. Li, X. Tang, and S. Panwar, “Modeling and analysis of an expiration-based hierarchical caching system,” in Proc. of 45th IEEE Globe Telecommunications Conf. (GLOBECOM02), pp. 2480-2484, 2002.

Hou, Y., J. Pan, C. Wang, and B. Li, “On prefetching in hierarchical caching systems,” in Proc. of 38th IEEE Int. Conf. on Communications (ICC03), Anchorage, USA, pp. 814-818, May 11-15, 2003.

Hou, Y., J. Pan, K. Sohraby and X. Shen, “Coping miss synchronization in hierarchical caching systems with nonlinear TTL functions,” in Proc. of 39th IEEE Int. Conf. on Communications (ICC04), Paris, France, pp. 2194-2198, June 20-24, 2004.

Hou, Y., Y. Shi, and J. Pan, “A lifetime-aware flow routing algorithm for energy-constrained wireless sensor networks,” in Proc. of 22nd IEEE Military Communications Conf. (MILCOM03), Boston, MA, USA, pp. 603-608, Oct. 13-16, 2003.

Hou, Y., Y. Shi, J. Pan, and S. Midkiff, “Lifetime-optimal data routing in wireless sensor networks without flow splitting,” in Proc. of 1st IEEE Workshop on Broadband Advanced Sensor Networks (Basenets04), invited paper, San Jose, CA, USA, Oct. 25, 2004.

Hou, Y., Y. Shi, J. Pan, and S. Midkiff, “Optimal single-session flow routing for wireless sensor networks,” in Proc. of 58th IEEE Semiannual Vehicular Technology Conf . (VTC03), Orlando, FL, USA, pp. 2855-2859, Oct. 6-9, 2003.

Hou, Y., Y. Shi, J. Pan, and S. Midkiff, “Single-beam flow routing for wireless sensor networks,” in Proc. of 48th IEEE Globe Telecommunications Conf. (GLOBECOM05), St. Louis, MO, USA, pp. 3263-3268, Nov. 28-Dec. 2, 2005.

Hu, N., R. Dannenberg, and G. Tzanetakis, “Polyphonic audio matching and alignment for music retrieval,” in Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA,) New Paltz, NY Oct. 2003

Impagliazzo, R., and B.M. Kapron, “Logics for reasoning about cryptographic constructions,” in Proc. 44th IEEE Symp. on Foundations of Computer Science, pp. 372-383, 2003.

Izquierdo, L., D. Damian, and D.M. German, “Towards understanding requirements management in a special case of global software development: A study of asynchronous communication in the Open Source Community,” in Proc. of Int. Workshop on Distributed Software Development, pp. 171-185, 2005.

Jahnke, J., D. German, and J. Wadsack, “Architectural patterns for data mediation in web-centric information systems,” in Proc. of the '02 ICSE Web Engineering Workshop, May 2002.

Jahnke, J., H.A. Müller, A. Walenstein, N. Mansurov, and K. Wong, “Fused data-centric visualizations for software evolution environments,” Proceedings IEEE 10th International Workshop on Program Comprehension (IWPC-2002), pp. 187-196, (Paris, France, June 26-29, 2002).

Jahnke, J.H., “Engineering component-based net-centric systems for embedded applications,” in Proc. of Joint European Software Engineering Conf. and Foundations of Software Engineering (ESEC/FSE '01), ACM Press, Vienna, pp. 218-228, Sep. 2001.

Jahnke, J.H., “Integrating cognitive support with existing case-tools for design recovery, “ in Proc. of the 2nd Workshop on Quantiative and Soft Computing Based Software Engineering (QSSE), 10 pages, Banff, Alberta, Feb. 2002.

Jahnke, J.H., “Reverse engineering software architecture using rough clusters,” in Fuzzy Information, Proc. of NAFIPS'04 ,IEEE Annual Meeting, vol. 1, no. 27-30, pp. 4-9, June 2004.

Jahnke, J.H., “Selecting a theory for managing imperfect knowledge in database reverse engineering processes,” in Proc. of IASTED Int. Conf. on Artificial Intelligence and Soft Computing (ASC 2001), 8 pages, Acta Press, 2001, Cancun, Mexico, May 2001.

Jahnke, J.H., “Toward context-aware computing in clinical care,” OOPSLA Workshop on Building Software for Pervasive Computing, San Diego, USA, 2005.

Jahnke, J.H., “Towards an architecture for refactoring embedded software for ubiquitous environments,” Workshop on Software Architecture Recovery and Modelling (SWARM'01), co-located with Working Conf. on Reverse Engineering (WCRE 2001), 6 pages Stuttgart, Germany, 2001.

Jahnke, J.H., A. Walenstein, "Reverse engineering tools as media for imperfect knowledge," in Proc. of the Working Conf. on Reverse Engineering (WCRE 2000), IEEE CS Press, Brisbane, Australia, pp. 22-31, 2000.

Jahnke, J.H., and J. Wadsack, "The varlet analyst: employing imperfect knowledge in database reverse engineering tools," in Proc. of the 3rd Int. Workshop on Intelligent Software Engineering (WISE-3), 10 pages, Limerick, Ireland, 2000.

Jahnke, J.H., and L. Kawasme, "Generation of asynchronous component adapters," Proc. of Int. ETAPS Workshop on Component Composition (SC'03), Electronic Notes in Theoretical Computer Science, 82 (5), 16 pages, Spinger, 2003.

Jahnke, J.H., and M. Dagpinar, "Predicting maintainability with object-oriented metrics - an empirical study, " Proc. of 10th Working Conf. on Reverse Engineering (WCRE), IEEE Computer Society, 10 pages, 2003.

Jahnke, J.H., and Y. Bychkov, "Reverse engineering software architecture using rough clusters," Proc. of 16th Int. Conf. on Software Engineering & Knowledge Engineering (SEKE'2004), ISBN 1-891706-14-4, Banff, Alberta, Canada, June 20-24, 2004.

Jahnke, J.H., I. Bilykh, B. Kursawe, Y. Bychkov, D. Dahlem, A. Onabajo, C. Kuziemsky, and F. Lau, "Society information grids," Proc. of Net. Objectdays 2003, Lecture Notes in Computer Science, Springer,14 pages, 2003.

Jahnke, J.H., J. Niere, and J. Wadsack, "Automated quality analysis of component software for embedded systems," in Proc. of the 8th Int. Workshop on Program Comprehension (IWPC 2000), IEEE CS Press, Limerick, Ireland, pp. 18-26, 2000.

Jahnke, J.H., J.P. Wadsack, "A history concept for design recovery tools," in Proc. of 6th Conf. on Software Maintenance and Reverse Engineering (CSMR 2002), Thermal Hotel, Margitsziget, Budapest, Hungary, pp. 37-46 Mar. 11-13, 2002.

Jahnke, J.H., K. Kontagiannis, E. Stroulia, S. Tilley, and K. Wong (eds.), "3rd Int. workshop on net-centric computing (NCC'01)," in Proc. of Int. Software Engineering Conf., Toronto, Ontario, pp. 766-767, 2001.

Jahnke, J.H., M. d'Entremont, M. Lavender, and A. McNair, "User-programming of net-centric embedded control software," in Proc. of Int. CHI Workshop on End-User Programming, pp. 39-43, 2003.

Jahnke, J.H., P. de Souza, and A. McNair, "Refactoring embedded to software into component-based architectures, " in Proc. of ASERC Workshop on Software Architecture (AWSA 2003), Banff, Alberta, 8 pages, Feb. 2003.

Jahnke, J.H., Y.B. Bychkov, D. Dahlem, L. Kawasme, "Context-aware information services for health care," Proc. of Workshop on Modelling and Retrieval of Contest, CEUR, ISSN 1613-0073, vol. 114, 2004.

Jahnke, J.H., Y.B. Bychkov, D. Dahlem, L. Kawasme, "Context-aware, mobile web services," OOPSLA 2004 Workshop on Building Software for Pervasive Computing,Vancouver, Canada, Oct. 2004.

Johnson, D.A., G.C. Shoja, "Request routing architecture in content delivery networks," Proc. of 2003 Int. Conf. on Internet Computing, Las Vegas, Nevaada, June 3-26, 2003.

Kabir, M.H., E.G. Manning and G.C.. Shoja, "Scalable multimedia streaming model and transmission scheme for VBR-encoded videos," Proc. 16th IASTED Int. Conf. on Parallel and Distributed Computing and Systems (PDCS2004), MIT, Cambridge, MA, USA, Nov. 9-11, 2004.

Kabir, M.H., E.G. Manning, and G.C. Shoja, "An interactive scalable multimedia streaming scheme for VBR-encoded videos," 3rd Annual Conf. on Communication Networks and Services Research (CNSR2005), Halifax, Nova Scotia, Canada, May 16-18, 2005.

Kabir, M.H., E.G. Manning, G.C. Shoja, "Request-routing trends and techniques in content distribution network," Proc. ICCIT 02, Dhaka, Bangladesh, pp. 315-320, ISBN 984-32-0450-6, Dec. 2002.

Kaminski, P., P. Agrawal, H. Kienle,and H.A. Müller, "I Need You! Initiative and Interaction in Autonomic Systems," Proceedings International Workshop on Design and Evolution of Autonomic Application Software (DEAS 2005), Workshop at 27th IEEE/ACM Int. Conf. on Software Engineering (ICSE 2005), St. Louis, MO; May 21, 2005, ACM Sheridan; ISBN 1-59593-025-6/05/0005; pp. 73-76, May 2005.

Kaminski, P., H.A. Müller, and M. Litoiu, "A Design for Adaptive Web Service Evolution," Proceedings International Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS

2006), Workshop at 28th IEEE/ACM Int. Conf. on Software Engineering (ICSE 2006), Shanghai, China; May 21-22, 2006, 7 pages, May 2006.

Kaminski, P. , H.A. Müller, and M. Litoiu, “A Design for Evolving Web Services,” Proceedings ACM IBM Center for Advanced Studies Conference (CASCON 2006); pp. 303-317; October 2006.

Kapur, A., A. Tindale, and G. Tzanetakis, “Learning indirect acquisition of instrumental gestures using direct sensors,” in Proc. of the IEEE Multimedia Signal Processing Workshop (MMSP), 2006.

Kapur, A., G. Tzanetakis, and P. Driessen, “Audio-based gesture extraction on the ESITAR controller,” in Proc. of the Int. Conf. on Digital Audio Effects (DAFX), Naples, Italy, 2004.

Kapur, A., G. Tzanetakis, N. Virji-Babul, G. Wang, and P. Cook, “A framework for sonification of motion capture data,” in Proc. of the Int. Conf. on Digital Audio Effects (DAFX), 2005.

Kapur, A., M. Benning, and G. Tzanetakis, “Query-by-beatboxing: music retrieval for the DJ,” in Proc. of the Int. Conf. on Music Information Retrieval (ISMIR), Barcelona, Spain, 2004.

Kapur, A., N. Virji-Babul, G. Tzanetakis, P. Driessen, “Gesture-based affective computing on motion capture data,” in Proc. of the Int. Conf. on Affective Computing and Intelligent Interaction (ACII), 2005.

Kapur, A., R. McWalter, and G. Tzanetakis, “New music interfaces for rhythm-based retrieval,” in Proc. of the Int. Conf. on Music Information Retrieval (SMIR), 2005.

Kent, K. B., J. C. Muzio, and G.C. Shoja, “Remote transparent execution of java threads,” Proc. of 2001 High Performance Computing Symp. (HPC'01), Seattle, Washington, U.S.A., pp. 184-191, Apr. 22-26, 2001.

Kent, K., and M. Serra, “Context switching in a hardware/software co-design of the Java virtual machine,” Design Automation and Test in Europe Conf., 2002.

Kent, K., and M. Serra, “Hardware architecture for Java in a hardware/software co-design of the virtual machine,” Euromicro Symp. on Digital System Design (DSD), 2002.

Kent, K., and M. Serra, “Hardware/Software Co-Design of a Java Virtual Machine,” IEEE Int. Workshop on Rapid Systems Prototyping (RSP), Paris, France, Jun. 2000.

Kent, K., and M. Serra, “Reconfigurable architecture requirements for co-designed virtual machines,” 10th IEEE Reconfigurable Architectures Workshop, part of IPDPS 2003, 2003.

Kent, K., and M. Serra, “The co-design of virtual machines using reconfigurable hardware,” Design Automation and Test Europe (DATE), PhD Forum, 2004.

Kent, K., H. Ma, and M. Serra, “ Rapid prototyping a co-designed Java Virtual Machine,” 15th Int. Workshop on Rapid System Prototyping (RSP), pp. 164-171, 2004.

Kerr, S., and D.M. German, “Partitioning the navigational model: a component-driven approach,” in Proc. of the Int. Conf. on web Engineering, July 2003.

Kienle, H., A. Weber, J. Jahnke, and H.A. Müller, “Tackling the adoption problem of domain-specific visual languages,” 2nd OOPSLA Workshop on Domain-Specific Visual Languages (DSVL 2002), 12 pages, (Seattle, USA, Nov. 4, 2002).

Kienle, H., A. Weber, J.H. Jahnke, and H.A. Müller, “Tackling the adoption problem of domain-specific visual languages,” in Proc. of OOPSLA 2002 Workshop on Domain-Specific Visual Languages, Seattle, 12 pages, Nov. 2002.

Kienle, H., and J.H. Jahnke, “A visual language with visio: first experiences,” in Proc. of 3rd Int. Workshop on Adoption-Centric Software Engineering (co-located with ICSE), Portland, 10 pages, May 2003.

Kienle, H.M., D.M. German, and H.A. Müller, “Legal concerns of web site reverse engineering,” in Proc. of the Int. Workshop in Web Site Evolution, pp. 41-50, Sept. 2004.

Kienle, H.M., D.M. German, S. Tilley, and H.A. Müller, “Intellectual property aspects of web publishing,” in SIGDOC '04: Proc. of the 22 Annual Int. Conf. on design of Communication, ACM Press, pp. 136-144, 2004.

Kienle, H., and H.A. Müller, “Leveraging program analysis for web site reverse engineering,” Proceedings 3rd IEEE International Workshop on Web Site Evolution (WSE 2001), pp. 117-125, (Florence, Italy, November 2001).

Kienle, H.M., and H.A. Müller, “A Lightweight Taxonomy to Characterize Component-Based Systems,” Sixth IEEE International Conference on COTS-based Software Systems (ICCBSS 2007), Banff, Alberta, Canada; pp. 192-201, February 26 - March 1, 2007.

King, V., and J. Saia, “Choosing a random peer,” Proc. of the 23rd Annual ACM Principles of Distributed Computing (PODC), ACM, pp. 115-124, July 2004.

King, V., and M. Thorup, “Space-saving trick for maintaining paths in dynamic reachability and shortest path algorithms,” in Proc. of the 7th Annual Int. Conf. on Computing and Combinatorics (COCOON), 2108, Springer, Aug. 2001.

King, V., J. Saia, V. Sanwalani, and E. Vee, “Towards secure and scalable computation in peer-to-peer networks,” in Proc. of 47th IEEE Symp. on Foundations of Computer Science (FOCS), Oct. 2006.

King, V., J. Saia, V. Sanwalani, E. Vee, “Scalable leader electron,” Proc. of 17th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA), Jan. 2006.

King, V., L. Zhang, and Y. Zhou, “On the complexity of reconstructing evolutionary trees,” in Proc. of the 14th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA), Jan. 2003.

King,V., O. Kupferman, and M. Vardi, “On the complexity of parity word automata,” in Foundations of Software Science and Technology (FOSSACS), April 2001.

Knuth, D.E., and F. Ruskey, “Efficient coroutine generation of constrained Gray sequences,” in From Object-Orientation to Formal Methods: Essays in Memory of Ole-Johan Dahl, eds. Olaf Owe et al., Lecture Notes in Computer Science vol. 2635, Springer-Verlag, pp. 183-208, 2004.

Kontogiannis, K. , G.A. Lewis, D.B. Smith, M. Litoiu, H.A. Müller, S. Schuster, and E. Stroulia, “The Landscape of Service-Oriented Systems: A Research Perspective,” Proceedings International Workshop on Systems Development in SOA Environments (SDSOA 2007); Workshop at 29th IEEE/ACM Int. Conf. on Software Engineering (ICSE 2007), Minneapolis, Minnesota, USA; May 21, 2007, 6 pages, May 2007.

Kountouriotis, V., P. Rondogiannis, and W.W. Wadge, “Extensional higher-order datalog,” Proc. of 12th Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-12), Jamaica, Dec. 2005.

Krall, A., N. Bermudo, and R.N. Horspool, “Control flow graph reconstruction for assembly language programs with delayed instructions,” Proc. of 5th Int. Workshop on Source Code Analysis and Manipulation, SCAM'05, Budapest, Hungary, IEEE Computer Society, pp. 107-116, Sept./Oct. 2005.

Kuziemsky, C.E., F. Lau, I. Bilykh, J.H. Jahnke, G. McCallum, C. Obry, A. Onabajo, and G.M. Downing, “Ontology-based information integration in health care: a focus on palliative care, ” Proc. of 11th Int. Conf. Software Technology and Engineering Practice (STEP), IEEE Computer Society, pp. 164-172, Amsterdam, Sep. 2004.

Kwan, I. and D. Damian, “Connecting people in a requirements-centred social network with requirements explorer,” Proc. of Workshop on Supporting the Socials-side of Software Development, Conf. on Computer-supported cooperative work (CSCW).

Kwan, I., D. Damian, and M.-A. Storey, “Visualizing a requirements-centred social network to maintain awareness within development teams,” Proc. of the Requirements Engineering Visualization Workshop, Int. Conf. on Requirements Engineering (RE06), Sep. 2006, to appear.

Lagrange, M., and G. Tzanetakis, “Sound source tracking and formation using normalized cuts,” in Proc. of the Int. Conf. on Audio, Speech & Signal Processing (ICASSP), 2007.

Lagrange, M., J. Murdoch, and G. Tzanetakis, “Temporal constraints for sound source formation using the normalized cut,” Neural Information Processing Systems (NIPS) Workship on Advances in Models for Acoustic Processing, 2006.

Lei, H., and G.C. Shoja, “A distributed trust model for e-commerce applications,” Proc. of 2005 IEEE Int. Conf. on e-Technology, e-Commerce, and e-Service, Hong Kong, pp. 290-293, Mar. 29-April 1, 2005.

Lei, H., G.C. Shoja, “Dynamic distributed trust model to control access to resources over the Internet,” Proc. of IEEE Pacific Rim Conf. on Computers, Communications and Signal Processing, pp. 988-991, Aug. 29-30, 2003.

Leveson, N.G., P. Allen and M.-A. Storey, “The Analysis of a Friendly Fire Accident using a Systems Model of Accidents,” 20th International System Safety Conference, Denver, Colorado, USA, Aug. 5-9, 2002.

Li, F. and K. Wu, “Fast and consistent packet classification scheme via efficient hash-caching,” Proc. of IASTED Int. Conf. on Communications and Computer Networks (CCN 02), Boston, pp. 420-425, Nov. 2002.

Li, F., K. Wu, and A. Lippman, “Energy efficient collaborative routing in multi-hop wireless ad hoc networks,” Proc. of 25th IEEE Int. Performance Computing and Communications Conf. (IPCCC 06), Phoenix Arizona, Apr. 2006.

Li, F., A. Lippman, and K. Wu, “Minimum-energy cooperative path routing in wireless networks: an integer programming formulation,” Proc. of 63th IEEE Vehicular Technology Conf. (VTC), Melbourne, Australia, May 2006.

Li, F., and K. Wu, “A power-efficient protocol for the construction of broadcast trees in multi-hop wireless ad hoc networks,” Proc. of IASTED Int. Conf. on Communications and Computer Networks (CCN 02), Boston, pp. 31-35, Nov. 2002.

Li, F., K. Wu, “Reliable, distributed, and power-efficient broadcasting in multi-hop mobile ad hoc networks,” Proc. of IEEE Int. Conf. on Local Computer Networks (LCN 02) and Workshop on Wireless Local Networks (WLN 02), Tampa, pp. 761-769.

Li, H., and W.W. Wadge, “An intensional tool applied to French language educational software,” SEKE, 2004.

Li, T., and G. Tzanetakis, “Factors in automatic musical genre classification of audio signals,” in Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, Oct. 2003

Li, Y. S. Bellamy, P. Driessen, and G. Tzanetakis, “Spatial sound rendering using measured room impulse responses,” in Proc. of the IEEE Int. Symp. on Signal Processing of Information Technology (ISSPIT), 2006.

Lintern R., J. Michaud, M.-A. Storey, and X. Wu, “Plugging-in Visualization: Experiences Integrating a Visualization Tool with Eclipse,” ACM Symp. on Software Visualization, (Softvis'2003), San Diego, Jun. 2003.

Lintern, R., and M.-A. Storey, “Jambalaya express: on demand knowledge visualization,” 8th Protégé Conf., Madrid, Spain, Jul. 18-21, 2005 (extended abstract).

Lippens, S., et al, “A comparison of human and automatic musical genre classification,” in Proc. of the IEEE Int. Conf. on Audio, Speech and Signal Processing(ICASSP), Montreal, Canada, 2004.

Liu, C, K. Wu, and V. King, “Randomized coverage-preserving scheduling schemes for wireless sensor networks,” 4th IFIP Int. Conf. on Networking, Waterloo, Canada, pp. 956-967, May 2005.

Liu, C., K. Wu, “Performance evaluation of range-free localization methods for wireless sensor networks,” to appear in 24th IEEE Int. Performance Computing and Communications Conf. (IPCCC 05), Phoenix Arizona, April 2005.

Liu, C., K. Wu, and J. Pei, “An energy efficient data collection framework for wireless sensor networks by adaptive spatial sampling rate adjustment,” Proc. of 2nd Annual IEEE Communications Society Conf. on Sensor and Ad Hoc Communications and Networks (SECON 05), Santa Clara, California, Sep. 2005.

Liu, C., K. Wu, and M. Taso, “Energy efficient information collection with the ARIMA model in wireless sensor networks,” Proc. of IEEE GlobeCom 2005, St. Louis, Missouri, USA, Nov. 2005.

Liu, C., K. Wu, V. King, “Very low cost sensor localization for hostile environments,” IEEE Int. Conf. on Communications (ICC), May 2005.

Liu, R., C. Gibbs, and Y. Coady, “MADAPT: managed aspects for dynamic adaptation based on profiling techniques,” 3rd Workshop on Reflective and Adaptive Middleware, Oct. 2004.

Liu, R., C. Gibbs, and Y. Coady, “SONAR: system optimization and navigation with aspects at runtime,” Dynamic Aspects Workshop, held at the Int. Conf. on Aspect-Oriented Software Development (AOSD), Mar. 2005.

Liu, Y, E. Stroulia, K. Wong, and D. German, “Using CVS historical information to understanding how students develop software,” in 1st Int. Workshop in Mining Software Repositories, pp. 32-36, May 2004.

Lomov, V.A., G.C. Shoja, “Adaptive topology protocol for network game support,” Proc. of IEEE Pacific Rim Conf. on Computers, Communications and Signal Processing, pp. 21-24 Aug. 29-30, 2003.

Long, B., D. Hoffman, and P. Strooper, “A concurrency test tool for Java monitors,” in Proceedings 16th Int. Conf. on Automated Software Engineering (ASE 2001), pp. 421-425, IEEE Computer Society, Nov. 2001.

Mahoney, G., W. Myrvold, and G.C. Shoja, “Generic reliability trust model,” Proc. of 3rd Annual Conf. on Privacy, Security and Trust, A. Ghorbani, and S. Marsh (editors), 7 pages, Oct. 2005.

Manning, E.G., “Quality of Service in Metro Networks: Admission Controls and the Importance of SONET,” keynote address, Proc. Int. Conf. on Distributed Computing Systems (ICDCS 04), IEEE, Tokyo, Mar. 26, 2004.

Manning, E.G., M.S. Rahman, Mostofa Akbar, and G.C.. Shoja, “Distributed utility model for distributed multimedia server systems,” Proc. DASD'05 Spring Simulation Multiconference, San Diego, 2005.

Mark, J., J. Pan, and X. Shen, “Mobility support in hybrid wireless/IP networking,” in Proc. of 3rd IEEE Int. Conf. on 3G Wireless Communications (3Gwireless02), San Francisco, CA, USA, pp. 346-351, May 28-31, 2002.

Mark, J., X. Shen, Y. Zeng, and J. Pan, “TCP/IP over wireless/wireline networks,” in Proc. of 2nd IEEE Int. Conf. on 3G Wireless Communications (3Gwireless00), Silicon Valley, CA, USA, pp. 438-445, June 14-16, 2000.

Marsono, M., M. El-Kharashi, F. Gebali, S. Ganti, “Distributed layer-3 email classification for SPAM control,” IEEE CCECE, Ottawa, May 2006.

Martin, J., and H.A. Müller, “C to Java migration experiences,” Proceedings of the Fifth European Conference on Software Maintenance and Reengineering (CSMR-2002), IEEE Computer Society Press, pp. 143-153, (Budapest, Hungary, March 11-13, 2002).

Martin, J., and H.A. Müller, “Strategies for migration from C to Java,” Proceedings of the Fifth European Conference on Software Maintenance and Reengineering (CSMR-2001), IEEE Computer Society Press, pp. 200-209, (Lisbon, Portugal, March 14-16, 2001).

Maslov, D., C. Young, D.M. Miller and G.W. Dueck, “Quantum Circuit Simplification using Templates,” 2005 Conference on Design and Test Europe (DATE), Munich, Germany, March 2005, pp. 1208-1213.

Maslov, D., D.M. Miller and G.W. Dueck, “Templates for Reversible Logic Synthesis,” PACRIM-2005, pp. 609-612, Victoria, Canada, August 2005.

Maslov, D., G.W. Dueck, and D.M. Miller, “Fredkin/Toffoli Templates for Reversible Logic Synthesis,” International Conference on Computer Aided Design (ICCAD), San Jose, CA, November 2003, pp. 256-261.

Maslov, D., G.W. Dueck, and D.M. Miller, “Simplification of Toffoli Networks via Templates,” 16th Symposium on Integrated Circuits and System Design, Sao Paulo, Brazil, September 2003, pp. 53-58.

Matthews, C., Y. Coady, and J. Appavoo, “Portability events: a programming model for scalable system infrastructures,” Programming Languages and Operating Systems (PLOS)Workshop, held at the 12th Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Oct. 2006.

McDonald, J., P. Strooper, and D. Hoffman, “Tool support for generating passive C++ oracles from Object-Z specifications,” in Proc. of 10th Asia-Pacific Software Engineering Conf., IEEE Computer Society, pp. 322-331, 2003.

McKay, Brendan D., Alison Meynert, and W. Myrvold, “Counting small latin squares,” European Women in Mathematics Int. Workshop on Groups and Graphs, Varna, Bulgaria, pp. 67-72, Aug. 31-Sept. 6, 2002.

Miao, Z., D. Stefanescu, and A. Thomo, “Grid-aware evaluation of regular path queries on spatial networks,” Proc. of the IEEE 21st Int. Conf. on Advanced Information Networking and Applications (AINA07), IEEE, 2007.

Michaud J., M.-A. Storey, and H.A. Müller, “Integrating information sources for visualizing Java programs,” Proceedings of the International Conference of Software Maintenance (ICSM'2002), pp. 250-259, Florence, Italy, Nov. 2001.

Michaud, J., and M.-A. Storey, “Role of knowledge in software customization,” SEKE2003, 15th Int. Conf. on Software Engineering and Knowledge Engineering, Hotel Sofitel, San Francisco Bay, July 1 - Jul. 3, 2003

Michaud, J., M.-A. Storey, and H.A. Müller, “Integrating information sources for visualizing Java programs,” Proceedings IEEE International Conference on Software Maintenance (ICSM 2001), pp. 250-259, (Florence Italy, November 2001).

Miller, D.M. and M.A. Thornton, “QMDD: A Decision Diagram Structure for Reversible and Quantum Circuits,” Proc. 2006 Int. Symposium on Multiple-Valued Logic, Singapore, May 2006, CD, 6 pp.

Miller, D.M. and R. Drechsler, “Further Improvements in Implementing MVDDs,” 2002 Int. Symposium on Multiple-Valued Logic, pp. 245-253, May 2002.

Miller, D.M., “Spectral and Two-Place Decomposition Techniques in Reversible Logic,” Midwest Symposium on Circuits and Systems, published on CD-ROM, August 2002.

Miller, D.M., and G.W. Dueck, “On the Size of Multiple-Valued Decision Diagrams,” Proc. 2003 Int. Symposium on Multiple-Valued Logic, Tokyo, Japan, May 2003, pp. 235-240.

Miller, D.M., and R. Drechsler, “Augmented sifting of multiple-valued decision diagrams,” Proc. 2003 Int. Symposium on Multiple-Valued Logic, Tokyo, Japan, May 2003, pp. 275-382.

Miller, D.M., G. Dueck, and D. Maslov, “A Synthesis Method for MVL Reversible Logic,” Proc. 2004 Int. Symposium on Multiple-Valued Logic, Toronto, Canada, May 2004, pp. 74-80.

Miller, D.M., M.A. Thornton and D. Goodman, “A Decision Diagram Package for Reversible and Quantum Circuit Simulation,” Proc. IEEE World Congress on Computational Intelligence, July 2006, CD, 8 pp.

Mori, T., M. Uchida, R. Kawahara, J. Pan, and S. Goto, “Identifying elephant flows through periodically sampled packets,” in Proc. of 2nd ACM SIGCOMM and USENIX Internet Measurement Conf. (IMC04), Taormina, Sicily, Italy, pp. 115-120, Oct. 25-27, 2004.

Müller, H.A., A. Weber, and K. Wong, “Leveraging cognitive support and modern platforms for adoption-centric software engineering (ACRE 2003),” Proceedings of the 3rd International Workshop on Adoption-centric Software Engineering (ACRE 2003), Portland, Oregon, pp. 30-35, May 2003.

Müller, H.A., “Bits of History, Challenges for the Future and Autonomic Computing Technology,” 13th IEEE Working Conference on Reverse Engineering (WCRE 2006); Invited Keynote Paper; pp. 9-15, Benevento, Italy; October 2006.

Murdoch, J., and G. Tzanetakis, “Interactive content-aware music browsing using the radio drum,” in Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME), 2006.

Muzio, J.C., and M. Serra, “HCI challenges in designing for users with disabilities,” Proc. 9th Int. Conf. on Human-Computer Interaction, New Orleans, vol. 1, pp. 46-50, Aug. 2001.

Myers, D., E. Hargreaves, J. Ryall, S. Thompson, M. Burgess, D. German, and M.-A. Storey, “Developing marking support within eclipse,” Eclipse Technology Exchange and Workshop, 19th Annual ACM Conf. on Object-Oriented Programming, Systems, Languages, and Applications, Vancouver, Oct. 24-28, 2004.

Myers, D., E. Hargreaves, J. Ryall, S. Thompson, M. Burgess, D.M. German, and M.A. Storey, “Developing market support within eclipse,” in Proc. of OOPSLA Workshop on Eclipse Technology eXchange, Sept. 2004.

Nagata, Y., D.M. Miller and M. Mukaidono, “Logic Synthesis of Controllers for B-Ternary Asynchronous Systems,” Proc. 2000 Int. Symposium on Multiple-Valued Logic, pp. 402-407, May 2000.

Neti , S., and H.A. Müller, “Quality Criteria and an Analysis Framework for Self-Healing Systems,” Proc. Intl .Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2007); Workshop at 29th IEEE/ACM Int. Conf. on Software Engineering (ICSE 2007), Minneapolis, Minnesota, USA; May 26-27, 2007, 10 pages, May 2007.

Netisopakul, P., L. White, D. Hoffman, and J. Morris, “Data coverage testing of programs for container classes,” in Proc. of 12th Int. Symp. on Software Reliability Engineering, pp. 183 -194, IEEE Computer Society, Nov. 2002.

Nikolic, M., D. Hoffman, and L. Trajkovic, “Mbone webcast: network setup and data collection,” in Proc. of 7th IASTED Int. Conf. on Internet Multimedia Systems and Applications, pp. 140-145, Aug. 2003.

Nomikos, C., P. Rondogiannis, and W.W. Wadge, “A sufficient condition for strong equivalence under the well-founded semantics,” in Proc. of 21st Int. Conf. on Logic Programming (ICLP 2005), Sitges, Spain, pp. 414-415, Oct. 2005.

Norris, C., and D.M. Miller, International Conference on Communications in Computing (CIC 2002).

Obry, C., and J.H. Weber-Jahnke, “Managing privacy in peer-to peer distribution of clinical documents,” Int. Conf. on Information Technology and Communication in Health, Victoria, B.C., Canada, 2007.

Obry, C., J.H. Jahnke, A. Onabajo, and W. Schäfer, “Enabling privacy in cross-organizational information mediation - an application in health care,” Proc. of the 11th Int. Conf. on Software Technology and Engineering Practice (STEP), IEEE Computer Society, Software Technology and Engineering Practice (STEP), L. O'Brien, N. Gold, K. Kontogiannis (eds.), pp. 155-163, Amsterdam, Sep. 2004.

Onabajo, A., and J.H. Jahnke, “Privacy requirements for assistive living infrastructure,” Int. Conf. on Information Technology and Communication in Health, Victoria, B.C., 2007.

Onabajo, A., and J.H. Jahnke, “Modelling and reasoning for confidentiality requirements in software development,” Proc. of IEEE Conf. on Engineering of Computer Based Systems (ECBS 06), Potsdam, Germany, Mar. 2006.

Onabajo, A., and J.H. Jahnke,”Properties of confidentiality requirements,” IEEE Conf. on Computer-Based Medical Systems (CBMS), Utah, USA, Jun. 2006.

Onabajo, A., C. Obry, and J.H. Jahnke, “A policy-based security mechanism for distributed health networks, “ NSF/NSA/AFRL Workshop on Secure Knowledge Management, Buffalo, NY, Sep. 23-24, 2004.

Onabajo, A., I. Bilykh, and J.H. Jahnke, “Wrapping legacy medical systems for intergrated health network,” Proc. of NODe Workshop on Migration and Evolvability of Long-life Software Systems, 10 pages, 2003.

Pack, S., X. Shen, J. Mark, and J. Pan, “A comparative study of mobility management schemes in mobile hotspots,” in Proc. of 8th IEEE Wireless Communications and Networking Conf. (WCNC07), Hong Kong, China, Mar.11-15, 2007.

Pan, J., J. Mark, and X. Shen, “End-to-end TCP performance with local retransmissions,” in Proc. of 2nd Int. Conf. on Internet Computing (IC01), Las Vegas, NV, USA, pp. 332-338, June 25-28, 2001.

Pan, J., J. Mark, and X. Shen, “Improving performance of HTTP traffic over wireless links,” in Proc. of 1st Int. Seminar on Teletraffic and Network (ISTN00), Hangzhou, China, pp. 26-29, Nov. 15-17, 2000.

Pan, J., J. Mark, and X. Shen, “TCP performance and its improvement over wireless links,” in Proc. of IEEE 43rd IEEE Globe Telecommunications Conf. (GLOBECOM00), San Francisco, CA, USA, pp. 62-66, Nov. 28-30, 2000.

Pan, J., Y. Hou, L. Cai, Y. Shi, and X. Shen, “Topology control for wireless sensor networks,” in Proc. of 9th ACM Conf. on Mobile Computing and Networking (MOBICOM03), San Diego, CA, USA, pp. 286-299, Sept. 14-19, 2003.

Pang, J.C., G.C.. Shoja, and E.G. Manning, “Providing Soft Real- time QoS Guarantees for Java Threads,” Proc. Joint ACM Java Grande - ISCOPE 2001 Conference, sponsored by ACM SIGPLAN, Stanford University, California, Jun, 2-4, 2001.

Pang, J.C., G.C.. Shoja, and E.G. Manning, “Supporting soft real-time tasks and QoS on the java platform,” presented in the Sixth Int. Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS'01), San Francisco, California, U.S.A., Apr. 23, 2001.

Price, M., and J.H. Jahnke, “From the BC Primary Care Informatics Lab: EGADSS! it is decision support and TAPAS a model for technology assisted paper practice,” Proc. of e-Health Conf., Victoria B.C., Apr. 2006.

Price, M., J.H. Jahnke, “EGADSS-evidence based guideline and decision support system,” 2005 Partnership Symp. of the Canadian Instutite for Health Information (CIHI), Victoria, B.C., Nov. 2005.

Pryanishnikov, I., A. Krall, and R.N. Horspool, “Pointer alignment analysis for processors with SIMD instructions,” Proc. of 5th Workshop on Media and Streaming Processors, MSP'03, San Diego, pp. 50-57, Dec. 2003.

Pu, J., E.G. Manning, and G.C. Shoja, “Routing reliability analysis of partially disjoint paths,” Proc. of the PACRIM '01 (the 2001 IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing), Victoria, B.C., pp. 79-82, Aug. 26-28, 2001.

Pu, J., E.G. Manning, and G.C. Shoja, and A. Srinivasan, “A New Algorithm to Compute Alternate Paths in Reliable OSPF (ROSPF), 2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '2001), Las Vegas, Nevada, ISBN 1-892512-66-1, vol 1, pp. 299-304, Jun. 2001.

Pu, J., E.G. Manning, G.C. Shoja, “Reliable routing in MPLS networks,” Proc. IASTED Int. Conf. on Communications and Computer Networks (CCN),Cambridge, Mass., USA, Oct. 4-6, 2002.

Pu, J., M. Akhbar, E. Gowland, G.C. Shoja, E.G. Manning, “SLA Admission Controller for Reliable MPLS Networks,” Proc. 21st IASTED Int. Conf. on Applied Informatics, Innsbruck, Austria, ISBN 0-88986-341-5. ISSN 1027-2666, pp. 630-635, Feb. 10-13, 2003.

Rayside D., M. Litiou, M.-A. Storey, and C. Best, “Integrating SHriMP with the IBM WebSphere studio workbench,” Proceedings of CASCON '2001, Toronto, Canada, pp. 79-93, Nov. 2001.

Rayside, D., M. Litiou, and M.-A. Storey, “ Visualizing flow diagrams with ShriMP,” OOPSLA 2001 Workshop on Software Visualization, Tampa, FL, Oct. 15, 2001.

Rice, J., and J.C. Muzio, “Antisymmetries in the realization of Boolean functions,” in Proc. of the Int. Symp. on Circuits and Systems (ISCAS 2002), Scottsdale, Arizona, USA, (CD ROM, paper no. 2666).

Rice, J., and J.C. Muzio, “Antisymmetries in the representation of Boolean and multivalved logic,” Proc. of 2005 Int. Symp. on Multiple-Valved Logic, pp. 270-275, 2005.

Rice, J., and J.C. Muzio, “On the use of autocorrelation coefficients in the identification of three level decompositions,” Proc. of Int. Workshop on Logic Synthesis, Laguna Beach, 2003.

Rice, J., and J.C. Muzio, “Properties of autocorrelation coefficients,” 2003 IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing, 2003.

Rice, J., and J.C. Muzio, “Two new methods for calculating autocorrelation coefficients,” Proc. 4th Int. Workshop on Boolean Problems, pp. 69-76, Sept. 2000.

Rice, J., and J.C. Muzio, “Use of the autocorrelation function in the classification of switching functions,” in Proc. of the Euromicro Symp. on Digital System Design: Architectures, Methods and Tools (DSD 2002), Dortmund, Germany, pp. 244-251.

Rigby, P.C., D. Cubranic, S. Thompson, D.M. German, and M.-A. Storey, “Gild: an open-source environment to teach programming to novices,” in Open Educational Symp. of 1st Int. Conf. on Open Source Systems, pp. 338-340, July 2005.

Rigby, P.C., D. Cubranic, S. Thompson, D.M. German, and M.-A. Storey, “The challenges of creating open source educational software: the Gild experience,” Open Educational Symp., 1st Int. Conf. on Open Source Systems, Genova, Italy, Jul. 11-15, 2005.

Rondogiannis, P., and W.W. Wadge, “A model intersection theorem for logic programming with negation-as-failure,” in Proc. of 4th Panhellenic Logic Symp., Thessaloniki, Greece, pp. 163-167, Jul. 2003.

Rondogiannis, P., and W.W. Wadge, “An infinite-game semantics for negation in logic programming, in Proc. of the Workshop on Games for Logic and Programming Languages (GaLoP), ETAPS 2005, Edinburgh, pp. 77-91, Apr. 2005.

Rondogiannis, P., and W.W. Wadge, “An infinite-valued semantics for logic programs with negation,” LNCS 2424, in Proc. of 8th European Conf. of Logics in Artificial Intelligence (JELIA2002), Cosenza, Italy, pp. 456-467, Sep. 2002.

Rondogiannis, P., and W.W. Wadge, “Minimum model semantics for logic programs with negation-as-failure, ACM Transactions on Computational Logic, vol. 6, no. 2, pp. 441-467, 2005.

Rondogiannis, P., and W.W.Wadge, “Game theory and well-founded negation,” 5th Panhellenic Logic Symp. (PLS5), Athens, Greece, Jul. 25-28, 2005.

Ruskey, F., and A. Williams, “Generating combinations by prefix shifts,” 11th Int. Computing and Combinatorics Conf. (COCOON), Kunming, China, Lecture Notes in Computer Science, vol. 3595, pp. 570-576, 2005.

Ruskey, F., and J. Sawada, “A fast algorithm to generate unlabelled necklaces,” 11th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA), pp. 256-262, 2000.

Ruskey, F., and J. Sawada, “Generating Necklaces and Strings with Forbidden Substrings,” 6th Annual Int. Combinatorics and Computing Conf. (COCOON), Lecture Notes in Computer Science, no.1858, pp. 330-339, July 2000.

Ruskey, F., and M. Weston, “More fun with symmetric Venn diagrams,” 3rd Int. Conf. on Fun with Algorithms, Isola d'Elba, Italy, pp. 235-246, 2004.

Ruskey, F., B. Miers, J. Sawada, and M. Serra, “The Number of Irreducible Polynomials over Finite Fields with Given Trace and Subtrace,” SIAM, 2000.

Scholz, B., E. Mehofer, and R.N. Horspool, “Partial redundancy elimination with prediction techniques,” Proc. of Euro-Par 2003, Klagenfurt, Austria, LNCS vol. 2790, Springer-Verlag, pp. 242-250, Aug. 2003.

Scholz, B., R.N. Horspool, and J. Knoop, “Optimizing for space and time usage with speculative partial redundancy elimination,” Proc. of LCTES 2004, Washington, DC, ACM Press, pp. 221-230, June 2004.

Scott, A., L. Izquierdo, S. Gupta, R. Elves, and D. Damian, “Leveraging design patterns in global software development,” Proc. of the Int. Workshop on Distributed Software Engineering, Paris, Aug. 2005.

Scott, A., U. Stege, and N. Zeh, “Politician's Firefighting,” 17th Int. Symp. on Algorithms and Computation (ISAAC' 06), in Kolkata, India, LNCS 4288, pp. 608-617, December 18-20, 2006.

Scott, T., K. Wu, and D. Hoffman, “Experimental study on the radio propagation pattern in wireless sensor networks,” in Int. Wireless Communications and Mobile Computing Conf. (IWCMC), Vancouver, 2006.

Scott, T., K. Wu, and D. Hoffman, “Radio propagation patterns in wireless sensor networks: new experimental results,” in Proc. Int. Symp. on Wireless Local and Personal Area Networks, Vancouver, July 2006.

Serra, M., and J.C. Muzio, “The IT support for acquired brain injury patients -the design and evaluation of a new software package,” Hawaii Int. Conf. on System Sciences - 35, Hawaii, Jan. 2002.

Serra, M., and K. Kent, “Using FPGAs to solve the Hamiltonian cycle problem,” ISCAS, 2003.

Serra, M., E. Wang, and J.C. Muzio, “Using multimedia in a digital design lab,” Proc. European Workshop on Microelectronics Education, May 2000, appearing in Microelectronics Education (ed. B. Courtois, N. Guillemot, G. Kamarinos and G. Stéhelin), Kluwer Academic Publishers, pp. 139-142, 2000.

Shabawa, K., and D.M. German, “BioFOSS: a survey of free/open source software in Bioinformatics,” 2006 IEEE Symp. on Computer Based Medical Systems, pp. 861-866, June 2006.

Shelford, S., M.M. Akbar, E.G. Manning and G.C. Shoja, “Distributed Optimal Admission Controllers for Service Level Agreements in Interconnected Networds,” Proc. 21st IASTED International Conference on Applied Informatics, Innsbruck, Austria, ISBN 0-88986-341-5 ISSN 1027-2666, PP, 565-570, Feb. 10-13, 2003.

Shpak, D.J., P.-C. Tang, and E.G. Manning, “Supporting Everything-on-demand in Cable Services,” Proc. 19th National Fiber Optics Engineers Conf., Orlando, Sep. 2003.

Sim, S.E., and M.-A. Storey, “A structured demonstration of program comprehension tools,” in Proc. of the 7th Working Conference on Reverse Engineering (WCRE '00), pp. 184-193, Nov. 2000.

Singer, J., R. Elves, and M.-A. Storey, “NavTracks: supporting navigation in software maintenance,” submitted to Int. Conf. on Software Maintenance (ICSM05), Budapest, pp. 325-334, Sep. 2005.

Singh, N., C. Gibbs, and Y. Coady, “C-CLR: a tool for configurable system software,” in Proc. of ACP4IS Workshop, held at the Int. Conf. on Aspect-Oriented Software Development (AOSD), Mar. 2007 (to appear).

Stamplee, O., C. Gibbs, and Y. Coady, “RADAR: really low level aspects for dynamic analysis and reasoning,” Workshop on Programming Languages and Operating Systems, held at the European Conf. on Object-Oriented Programming (ECOOP), June 2004.

Stefanescu, D., A. Thomo, and L. Thomo, “Distributed evaluation of generalized path queries,” Proc. of the 20th Annual ACM Symp. on Applied Computing (SAC05), ACM Press, 2005.

Stefanescu, D., and A. Thomo, “Enhanced regular path queries on semistructured databases,” Proc. of 11th Int. Workshop on Foundations of Models and Languages for Data and Objects (FMLDO06), Springer, 2006.

Stege, U., I. van Rooij, A. Hertel, and P. Hertel, “An O(pn+1.151p)-algorithm for p-profit cover and its practical implications for vertex cover,” Proc. of the 13th Int. Symp. on Algorithms and Computation (ISAAC'02), Vancouver, Canada, LNCS 2518, pp. 249-261, 2002.

Stier, J., J.H. Jahnke, and H. Müller, “Simulating mechatronic systems by geometry-driven Petri net,” Proc. of 4th Symp. on Design, Analysis, and Simulation of Distributed Systems, SCS, Huntsville, Alabama, USA, Apr. 2006.

Stier, J., J.H. Jahnke, H. Müller, “Geist3D, a simulation tool for geometry-driven Petri nets, Proc of 27th Int. Conf. on Application and Theory of Petri Nets and Other Models of Councurrency,” Springer Verlag, LNCS, Turku, Finland, Jun. 26-30, 2006.

Storey, M.-A., N.F. Noy, M. Musen, C. Best, and R. Fergerson, “Jambalaya: an interactive environment for exploring ontologies,” International Conference on Intelligent User Interfaces, San Francisco, California, pp. 239, Jan. 13-16, 2002 (short paper).

Storey, M., J. Michaud, M. Mindel, M., Sanseverino, D. Damian, and al., “Improving the usability of eclipse for novice programmers,” Proc. of the 2003 OOPSLA Workshop on Eclipse Technology Exchange, Conf. on Object Oriented Programming Systems Languages and Applications, Anaheim, pp. 35-39, 2003.

Storey, M.-A., C. Best, J. Michaud, D. Rayside, M. Litoiu, and M. Musen, “SHriMP views: an interactive environment for information visualization and navigation,” Proceedings of CHI 2002. Conference Extended Abstracts on Human Factors in Computer Systems, Minneapolis, Minnesota, USA, pp. 520-521, Apr. 20-25, 2002.

Storey, M.-A., “ShriMP views: an interactive environment for exploring multiple hierarchical views of a Java program,” ICSE 2001 Workshop on Software Visualization, Toronto, Ontario, Canada, May 12, 2001.

Storey, M.-A., D. Cubranic, and D.M. German, “On the use of visualization to support awareness of human activities in software development: a survey and a framework,” in Proc. of the 2nd ACM Symp. on Software Visualization, pp. 193-202, May 2005.

Storey, M.-A., J. Michaud, M. Mindel, M. Sanseverino, D. Damian, D. Myers, D. German, and E. Hargreaves, “Improving the usability of Eclipse for novice programmers,” in Proc. of the OOPSLA Workshop on Eclipse Technology eXchange, pp. 35-39, Oct. 2003.

Storey, M.-A., L.-T. Cheng, P. Rigby, and I. Bull, “Waypointing and Social Tagging to Support Program Navigation,” CHI Extended Abstracts, Apr. 22-26, 2006, Montreal, Canada.

Storey, M.-A., M. Musen, J. Silva, C. Best, N. Ernst, R. Fergerson, and N. Noy, “Jambalaya: interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé,” appeared in a workshop on Interactive Tools for Knowledge Capture, K-CAP-2001, Victoria, B.C., Canada, Oct. 20, 2001.

Storey, M.-A., M. Sanseverino, D. German, D. Damian, A. Damian, J. Michaud, A. Murray, R. Lintern, J. Chisan, M. Litoiu, and D. Rayside, “Adopting GILD: an integrated learning and development environment for programming,” ACSE 2003: 3rd Int. Workshop on Adoption-Centric Software Engineering ICSE 2003, the 25th Int. Conf. on Software Engineering, Portland, USA, May 2003.

Sun, B., C.C. Li, K. Wu, and Y. Xiao, “A LCG-based Secure Protocol for Wireless Sensor Networks,” Proc. of IEEE Int. Conf. on Communications (ICC), Istanbul, Turkey, Jun. 2006.

Sun, B., F. Yu, K. Wu, and V.C.M. Leung, “Mobility-based anomaly detection in cellular mobile networks,” Proc. of 3rd ACM Int. Workshop on Wireless Security (WiSe 04), in conjunction with ACM MOBICOM 2004, Philadelphia, PA, Oct. 2004.

Sun, B., K. Wu, and U.W. Pooch, “Alert aggregation in mobile ad hoc networks,” Proc. of 2nd ACM Int. Workshop on Wireless Security (WiSe 03), San Diego, California, pp. 69-78, Sep. 2003.

Sun, B., K. Wu, and U.W. Pooch, “Routing anomaly detection in mobile ad hoc networks,” Proc. of 12th Int. Conf. on Computer Communication and Networks (ICCCN 03), Dallas, Texas, pp. 25-31, Oct. 2003.

Sun, B., K. Wu, and U.W. Pooch, “Towards adaptive intrusion detection in mobile ad hoc networks,” Proc. of IEEE GlobeCom 2004, Dallas, Nov. 2004.

Sun, B., X. Jin, K. Wu, and Y. Xiao, “In-network aggregation and system monitoring for wireless sensor networks,” accepted by IEEE Int. Conf. on Communications (ICC), Glasgow, Scotland, June 2007.

Systä, T., P. Yu, and H.A. Müller, “Analyzing Java software by combining metrics and program visualization,” Proceedings of the Fourth European Conference on Software Maintenance and Reengineering (CSMR-2000), pp. 199-208, IEEE Computer Society Press, (Zürich, Switzerland, Feb. 29 - March 3, 2000).

Tang, A., M. Tory, B. Po, P. Neumann, and S. Carpendale, “Collaborative coupling over tabletop displays,” ACM CHI 2006, CHI Letters, pp. 1181-1190, 2006.

Thomo, A., and G. Grahne, “Query answering and containment for regular path queries under distortions,” Proc. of the 3rd Int. Symp. on Foundations of Information and Knowledge Systems (FoIKS04)}, Springer-Verlag, 2004.

Thornton, M., D.M. Miller and W. Townsend, “Chrestenson Spectrum Computation Using Cayley Color Graphs,” 2002 Int. Symposium on Multiple-Valued Logic, pp. 123-128, May 2002.

Thornton, M.A., R. Drechsler, and D.M. Miller, “Multi-output Timed Shannon Circuits,” ISVLSI 2002.

Thornton, M.A., R. Drechsler, D.M. Miller and W. Townsend, “Computing Walsh, Arithmetic and Reed-Muller Spectral Decision Diagrams Using Graph Transformations,” GLVLSI 2002.

Tindale, A., A. Kapur, G. Tzanetakis, and I. Fujinaga, “Retrieval of percussion gestures using timbre classification techniques,” in Proc. of the Int. Conf. on Music Information Retrieval (ISMIR), Barcelona, Spain, 2004.

Tindale, A., A. Kapur, G. Tzanetakis, P. Driessen, and W.A. Schloss, “A comparison of sensor strategies for capturing percussive gestures,” in Proc. of the New Interfaces for Music Expression (NIME), Vancouver, BC, May 2005.

Tindale, A., A. Kapur, W.A. Schloss, and G. Tzanetakis, “Indirect acquisition of percussive gestures using timbre recognition,” in Proc. of the Conf. on Interdisciplinary Musicology (CIM), Montreal, Canada, 2005.

Tindale, Al, A. Kapur, G. Tzanetakis, and P. Driessen, “A comparison of sensor strategies for capturing percussive gestures,” in Proc. of the Int. Conf. on New Interfaces for Musical Expression (NIME), 2005.

Tzanetakis, G., “Audio-based gender identification using bootstrapping,” in Proc. of the IEEE PacRim Conf., 2005.

Tzanetakis, G., “Musescape: a tool for changing music collections into libraries,” in Proc. of the European Conf. on Digital Libraries (ECDL), Trondheim, Norway, 2003.

Tzanetakis, G., “Musescape: an interactive content-aware music browser,” in Proc. of the Int. Conf. on Digital Audio Effects (DAFX), London, UK, 2003.

Tzanetakis, G., “Song-specific bootstrapping of singing voice structure,” Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME),Taipei, Taiwan, 2004.

Tzanetakis, G., A. Kapur, R. McWalter, “Subband-based drum transcription for audio signals,” in Proc. of the IEEE W orkshop on Multimedia Signal Processing (MMSP), 2005.

Tzanetakis, G., and M.Y. Chen, “Building audio classifiers for broadcast news retrieval,” in Proc. of the IEEE Workshop on Image Analysis for Multimedia Interactive Services, Lisbon, Portugal, 2004.

Tzanetakis, G., J. Gao, and P. Steenkiste, “A scalable peer-to-peer system for music content and information retrieval,” in Proc. of the Int. Conf. on Music Information Retrieval (ISMIR), Baltimore, US, 2003.

van Emden, M.H., “Combining numerical analysis and constraint processing by means of controlled propagation and redundant constraints,” First COCOS Workshop on Intervals and Optimization, Oct. 2002.

van Emden, M.H., “Functional and relational box consistency by structured propagation in atomic constraint systems,” Sixth Annual Workshop of the ERCIM Working Group on Constraints, Prague, Czechia, 2001.

Wadge, W.W., “IML III,” 2005 Int. Conf. on Programming Languages and Compilers, Las Vegas, Jun. 27-30, 2005.

Wadge, W. W., “'The medium is the message,” Proc. DCW 2002 (4th Int. Conf. on Distributed Communities on the Web), Sydney, Australia, 2002 (revised papers to appear in the LNCS series).

Wadge, W.W., “Intensional markup language,” Proc. DCW 2000 (Workshop on Distributed Communities on the Web), Quebec, Canada, June 2000 (LNCS 1830).

Wadsack, J., J.H. Jahnke, and D. German, “Architectural patterns for data mediation in web-centric information systems,” in Proc. of the Int. Workshop on Web Engineering at ICSE 2002, Orlando, 8 pages, May, 2002.

Wadsack, J., and J.H. Jahnke, “Towards model-driven middleware maintenance, “ in Proc. of OOPSLA 2002 Workshop on Model-Driven Architecture, Seattle, 10 pages, Nov. 2002.

Wadsack, J., J. Niere, H. Giese, and J.H. Jahnke, "Towards data dependency detection in web information systems," in Proc. on Database Maintenance and Reengineering Workshop, 6 pages, Montreal, Oct. 2002.

Wang, J.Y.Q., J.S. Ong, Y. Coady, and M.J. Feeley, "Using idle workstations to implement predictive prefetching," in Proc. of 9th IEEE Symp. on High Performance Distributed Computing (HPDC-9), pp. 87-94, Aug. 2000.

Warfield, A., Y. Coady, and N. Hutchinson, "Identifying open problems in distributed systems," in Proc. of European Research Seminar on Advances in Distributed Systems (ERSADS), May 2001.

Watson, R.K., S. Khan, K. Li, E.G. Manning and M. Akbar, "Optimal Admission Controllers for Service Level Agreements in Enterprise Networks," Proc. SCI 2002, Sixth World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Fla., 14 Jul. 2002

Weber, A., H. Kienle, and H.A. Müller, "Live documents with contextual, data-driven information components," Proceedings of SIGDOC 2002, 10 pages, (Toronto, Canada, Oct. 20-23, 2002).

Weber-Jahnke, J.H., "Harmony - A FUJABA plug-in for designing HL7 Messages," Proc. of Fujaba Days 2006 Conf., Bayreuth, Germany, p.61-67, 2006.

Weber-Jahnke, J.H., G. McCallum, and M. Price, "Making available clinical decision support in service-oriented architectures," Int. Conf. on Information Technology and Communication in Health, Victoria, B.C., 2007.

Wu, J., and M.-A. Storey, "A multi-perspective software visualization environment," in Proc. of CASCON'2000, pp. 41-50, Nov. 2000.

Wu, K., C. Liu, and V. King, "Very low cost sensor localization for hostile environments," to appear in IEEE ICC, Seoul, Korea, May 2005.

Wu, K., C. Liu, Y. Xiao, and J.C. Liu, "Coarse-grained scheduling for gateway nodes in wireless sensor networks," accepted by IEEE Wireless Communications and Networking Conf. (WCNC) as a regular paper, 2007.

Wu, K., H.C. Yang, and F. Li, "Pessimism is mostly the best in the expanding ring search for wireless networks," Proc. of IEEE GlobeCom 2006.

Wu, X., A. Murray, M.-A. Storey, and R. Lintern, "A reverse engineering approach to support software maintenance: version control knowledge extraction," 11th IEEE Working Conf. on Reverse Engineering (WCRE 2004), Delft, the Netherlands, pp. 90-99, Nov. 9-12, 2004.

Xiao, Y. H. Chen, K. Wu, C. Liu, and B. Sun, "Maximizing network lifetime under QoS constraints in wireless sensor networks," Proc. of IEEE GlobeCom 2006.

Xiao, Y., H. Chen, K. Wu, B. Sun, and C. Liu, "Modeling detection metrics in randomized scheduling algorithm in wireless sensor networks," accepted by IEEE Wireless Communications and Networking Conf. (WCNC) as a regular paper, 2007.

Xiao, Y., and K. Wu, "Location update for PCS networks with a fractional movement threshold," MWN-3003 Workshop on Mobile and Wireless Networks, Providence, Rhode Island, May 2003.

Xiao, Y., H. Li, K. Wu, K. Leung, and Q. Ni, "Reservation with grouping stations for the IEEE 802.11 DCF," to appear in 4th IFIP Int. Conf. on Networking, Waterloo, Canada, May 2005.

Yang, L., H.C. Yang, and K. Wu, "Minimum-energy routing configuration for wireless ad hoc networks," Proc. of 25th IEEE Int. Performance Computing and Communications Conf. (IPCCC 06), Phoenix Arizona, Apr. 2006.

Yang, L.T., and J.C. Muzio, "An improved register transfer level built-in self-test partitioning," Proc. 9th Int. Symp. on Integrated Circuits, Devices and Systems, Singapore, pp. 414-417, Sept. 2001.

Yang, L.T., and J.C. Muzio, "High-level data path synthesis for built-in self-test designs," Proc. 2001 IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing, Victoria, pp. 279-282, Aug. 2001.

Yang, L.T., and J. Muzio, "An integrated high-level test synthesis algorithm for built-in self-testable designs," in Proc. of the XIV Int. Symp. on Integrated Circuits and System Designs (SBCCI-01), Brasilia, Brazil, pp. 115-121, Sept. 10-15, 2001.

Yang, L.T., and J. Muzio, "Redundant transformations for the testability metrics-based built-in self-testable data path allocation," in Proc. of the 2002 IEEE Asia-Pacific Conf. on Circuits and Systems (APCCAS-02), Bali, Indonesia, vol. 2, pp. 119-123, Oct. 28-31, 2002.

Yang, L.T., and J.C. Muzio, “A BIST testability metric-based algorithm to integrate scheduling and allocation in high level test synthesis,” Proc. 9th Int. Symp. on Integrated Circuits, Devices and Systems, Singapore, pp. 409-413, Sept. 2001.

Yang, L.T., and J.C. Muzio, “A high-level data path allocation algorithm based on BIST testability metrics,” in Proc. of the 14th Int. Conf. On Microelectronics (ICM-02), Beirut, Lebanon, Dec. 11-13, 2002.

Yang, L.T., and J.C. Muzio, “A register-transfer level BIST partitioning approach for ASIC designs,” Proc. 2001 IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing, Victoria, B.C., pp. 275-278, Aug. 2001.

Yang, L.T., and J.C. Muzio, “An improved BIST testability metric-based high-level test synthesis approach,” in Proc. of the 2002 Int. Conf. on VLSI (VLSI-02), Las Vegas, USA, pp. 78-85, June 24-27, 2002.

Yang, L.T., and J.C. Muzio, “An improved high-level built-in self-test synthesis algorithm,” Proc. 8th IEEE Int. Conf. on Electronic Circuits and Systems, Malta, vol. 1, pp. 549-552, Sept. 2001.

Yang, L.T., and J.C. Muzio, “Built-in self-testable data path synthesis,” Proc. 2001 IEEE Computer Society Annual Workshop on VlSI (WVLSI 2001), Orlando, pp. 78-84, Apr. 2001.

Yang, L.T., and J.C. Muzio, “High-level test synthesis for built-in self-testable designs,” Proc. XIV Symp. on Integrated Circuits and System Design, Brazil, pp. 115-121, Sept. 2001.

Yang, L.T., and J.C. Muzio, “Introducing redundant transformations for built-in self-testable data path allocation,” in Proc. of the 2002 IEEE Int. Conf. On Communications, Circuits and Systems (ICCCAS-02), Chengdu, China, vol. 2, pp. 1346-1350, June 29 - Jul.1, 2002.

Yang, L.T., and J.C. Muzio, “Introducing redundant transformations for high-level built-in self-testable synthesis,” in Proc. of the 2002 IEEE Int. Conf. on Electronics, Circuits, and Systems (ICECS-02), Dubrovnik, Croatia, vol. 2, pp. 475-479, Sep. 15-18, 2002.

Yang, L.T., and J.C. Muzio, “Redundant transformations for the testability metrics-based built-in self-testable data path allocation,” in Proc. of the 2002 IEEE Asia-Pacific Conf. on Circuits and Systems (APCCAS-02), Bali, Indonesia, Oct. 28-31, 2002.

Yang, L.T., and J.C. Muzio, “Redundant transformations for the testability metrics-based high-level built-in self-testable synthesis,” in Proc. of the XVII Int. Conf. on Design of Circuits and Integrated Systems (DCIS-02), Santander, Spain, Nov. 19-22, 2002.

Yeow, K., R.N. Horspool, and M.R. Levy, “Expeditious XML processing,” to appear in Proc. of IDEAL '02, 3rd Int. Conf. on Intelligent Data Engineering and Automated Learning, Manchester, UK, Springer-Verlag, LNCS vol. 2412, pp. 148-153, Aug. 2002.

Yu, L., G.C. Shoja, H.A. Müller, Anand Srinivasan, “A framework for live software upgrade,” Proc. of ISSRE 2002, Annapolis, MD, USA, pp. 149-158, Nov. 2002.

Yu, P., T. Systä, and H.A. Müller, “Predicting fault-proneness using OO metrics: an industrial case study,” Proceedings of the Fifth European Conference on Software Maintenance and Reengineering (CSMR-2002), IEEE Computer Society Press, pp. 99-107, (Budapest, Hungary, March 11-13, 2002).

Zastre, M., “Blogs and oral exams: engaging students in an 'IT & Society' course,” in Proc. of Western Canadian Conf. on Computing Education, May 2006.

Zastre, M., and R.N. Horspool, “Two techniques for improving the performance of exception handling,” Proc. of Workshop on Exception Handling in Object Oriented Systems, Glasgow, Scotland, Published as Tech. Report 05-050, Department of Computer Science, LIRMM, Montpellier-II University, pp. 63-74, July 2005.

Zhong, J. and J.C. Muzio, “An investigation of non-linear machines as PRPGs in BIST,” Proc. of 2004 Int. Conf. on ULSI, 2004.

Zhong, J. and J.C. Muzio, “Analyzing fault models for reversible logic circuits,” IEEE Congress on Evolutionary Computing, CEC 7133 on IEEE DVD 06CH37726D, July 2006.

Zhong, J. and J.C. Muzio, “Using crosspoint faults in simplifying Toffoli networks,” Proc. of 4th Int. IEEE NEWCAS Conf., June, 2006.

Zhong, J., and J.C. Muzio, “A comparison of generators for testing sequential circuits using BIST,” Proc. of Int. Workshop on Logic Synthesis, Laguna Beach, 2003.

Zhou, Z., X. Xiang, X. Wang, and J. Pan, “An energy-efficient data-dissemination protocol in wireless sensor networks,” in Proc. of 2006 IEEE Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM06), Buffalo, NY, USA, pp, 13-22, June 26-29, 2006.

Zowghi, D., and D. Damian, “Field studies of requirements engineering in a multi-site software development organization: research in progress,” Proc. of Australian Workshop on Requirements Engineering, Sydney, Australia, Nov. 2001.

Appendix C: Papers Published with Undergraduates

The names of undergraduate students in the lists of co-authors are shown in *italics*.

- Brendan D. McKay, Wendy Myrvold, and *Alison Meynert*. Small Latin Squares, Quasigroups and Loops. Journal of Combinatorial Designs. Accepted Nov. 9, 2005, 26 pages.
- P. W. Fowler, *D. Horspool*, and W. Myrvold. Vertex spirals in fullerenes and their implications for nomenclature of fullerene derivatives. Accepted to Chemistry: A European Journal, Nov. 2006, 11 pages.
- W. Myrvold, B. Bultena, S. Daugherty, B. Debroni, *S. Girn*, and *M. Minchenko*. FuiGui: A graphical user interface for investigating conjectures about fullerenes. Accepted to MATCH, Nov. 2006, 20 pages.
- Kapur Ajay, *McWalter Richard* and Tzanetakis George. "Subband-based Drum Transcription of Audio Signals" / IEEE Workshop on Multimedia Signal Processing (MMSP) / Shanghai, China, 2005.
- Kapur Ajay, *McWalter Richard* and Tzanetakis George "New Interfaces for Rhythm-based Retrieval" / Intl. Conf. on Music Information Retrieval (ISMIR) / London, UK, 2005.
- *Bray Stuart*, Tzanetakis George "Distributed Audio Feature Extraction for Music" / Intl. Conf. on Music Information Retrieval (ISMIR) / London, UK, 2005.
- *Bray Stuart*, Tzanetakis George. "Implicit Patching for Dataflow-based Audio Analysis and Synthesis" / Intl. Computer Music Conference (ICMC) / Barcelona, Spain, 2005
- Kapur Ajay, *Benning Manj*, and Tzanetakis George. "Query-by-Beat-Boxing: Music Retrieval for the DJ" / In Proc. Int.Conf. on Music Information Retrieval (ISMIR) / Barcelona, Spain, 2004.
- *B. D. Bingham*, Dale D. Olesky and P. van den Driessche, "Potentially nilpotent and spectrally arbitrary even cycle sign patterns", to appear in Linear Algebra and its Applications.
- Dale D. Olesky, *Aidan Roy* and P. van den Driessche, "Maximal graphs and graphs with maximal spectral radius", Linear Algebra and its Applications, 346, pp. 109-130, 2002.
- V. Mehrmann, Dale D. Olesky, *T.X.T. Phan* and P. van den Driessche, "Relations between Perron-Frobenius results for matrix pencils", Linear Algebra and its Applications, 287, pp. 257-269, 1999.
- *Gibbs, C.*, and Y. Coady, "Garbage collection in jikes: could dynamic aspects add value?" in Proc. of 1st AOSD Aspects Workshop (DAW'04), held at the Int. Conf. on Aspect-Oriented Software Development, Mar. 2004.
- Stamplee, O., *C. Gibbs*, and Y. Coady, "RADAR: really low level aspects for dynamic analysis and reasoning," Workshop on Programming Languages and Operating Systems, held at the European Conf. on Object-Oriented Programming (ECOOP), June 2004.
- *Gibbs, C.*, and Y. Coady, "OASIS: organic aspects for system infrastructure software," Workshop on Reflection, AOP and Meta-Data for Software Evolution, (held at the European Conf. on Object-Oriented Programming (ECOOP), June 2004.

- Gibbs, C., *P. Proctor,* and Y. Coady, “Surrendering to the need for speed while maintaining visibility in adverse code conditions,” in Proc. of LATE Workshop, held at the Int. Conf. on Aspect-Oriented Software Development (AOSD), Mar. 2007 (to appear).
- Stege, U., I. van Rooij, *A. Hertel*, and *P. Hertel*, “An O(pn+1.151p)-algorithm for p-profit cover and its practical implications for vertex cover,” in Proc. of the 13th Int. Symp. on Algorithms and Computation (ISAAC’02), Vancouver, Canada, LNCS 2518, pp. 249-261, 2002.

This is a paper produced by an undergraduate without a faculty co-author, but where the problem was proposed by the faculty member (F. Ruskey) as an honours project.

- *A. D. King*. “Generating Indecomposable Permutations.” Discrete Mathematics, 306, pp. 508-518, 2006.

Appendix D: Involvement of Adjunct Professors

Ian Barrodale was the founding Chair of the department. He left to form a company, Barrodale Computer Systems, but has maintained a close relationship with the university and the department in particular. As an illustration of the typical involvement of the adjunct professors in our department, this is a list of students who have worked with him.

D.1 UVic Co-op Students and B. Sc. Graduates

- Wahiba Chair. Worked on developing a comprehensive testing system for the BCS Grid DataBlade product for handling large 4D grids.
- Wanpeng Chang. Assisted in the implementation of the IDM (INDEA Data Manager) - a GIS-like data management software system to support the Elections BC INDEA (Integrated Digital Electoral Atlas) database.
- Angus Christian. Performed data analysis and correction of INDEA spatial and attribute data using the IDM software system.
- Sarah Collins. Designed and implemented BCS web pages.
- Adon Cook. Assisted in system administration and in the development of a stream gradient estimation software system.
- Adrian Damian. Worked on the enhancement of the IDM.
- Sean Fadum. Performed data analysis and correction of INDEA spatial and attribute data using the INDEA data management software system.
- Wayne Fang. Assisted in the porting of the BCS Grid DataBlade from Informix to PostgreSQL and in the implementation of mountain pine beetle modeling capability in a timber supply modeling software system.
- Ryan Hofschneider. Worked on the implementation of indexing schemes for stream networks and in the development of the IDM and the BCS Grid DataBlade. Also assisted in systems administration.
- Ben Kerr. Performed data analysis and correction of INDEA spatial and attribute data using the IDM software system.
- Dennis Manke. Assisted in the design and development of an algorithm for generating watershed boundaries.
- Dennis Li. Assisted in the development of data correction software.
- Yiping Mao. Assisted in the development of INDEA database software.
- Mike Morley. Worked on the design and implementation of algorithms for matched field inversion of acoustic array data from the Gulf of Mexico gas hydrates monitoring station being developed by the University of Mississippi.
- Borna Noureddin. Assisted in the development of warping software using thin plate splines and in the development of the IDM software system. Also assisted in systems administration.
- Xinghai (Steven) Shao. Worked on the development of a query/batch processing tool to support INDEA workflow.
- Erin Wagg. Performed office management duties.

- Jack Wong. Assisted in the development of software for interactive display of hydrography data for the TRIM Watershed Atlas project.
- Ming Woon. Performed data analysis and correction of INDEA spatial and attribute data using the IDM software system.
- Jamie Workman. Assisted in the development of software for computing watershed boundaries for new hydrographic data while respecting previously computed boundaries.

D.2 Masters Graduates

- Bette Bultena. Developed an ARC/View based system for producing the Voter Area maps series for Elections BC.
- Huan-Huan (Echo) Liang. Assisted in the development of testing software for the Grid DataBlade and in the addition of functionality to the IDM.
- Kasia Muldner. Developed software for evaluating spatial predicates and for processing watershed data.
- Neil Walker. Assisted in the development of the IDM software system.

D.3 Ph. D. Graduates

- Carlos Escalante. Assisted in the analysis and processing of hydrographic data and watershed generation algorithms.

Appendix E: Combined Programs

E.1 History of Combined Programs

The following list summarizes the history of the Combined Degree programs offered through the Department of Computer Science. It is clear that there has been a surge in the introduction of these programs over the past few years. In order to provide some context, the dates of some traditional CSc programs are also included.

1978	BSc Computer Science (Major, Honours, General)
1979	BSc Computer Science (Co-op, Major and Honours)
1979	BSc Combined Math and CSc (Major and Honours), Fac. of Science
1979	Minor in Computer Science
1995	BSc Combined Statistics and CSc (Major and Honours), Fac. of Science
1998	BSc Combined Physics and CSc (Major and Honours), Fac. of Science
2003	BSc Combined Visual Arts and CSc (Major), Fac. of Fine Arts
2003	BSc Combined Health and Information Science and CSc (Co-op, Major), HSDA
2004	B.A. Combined Psychology and CSc (Major), Fac. of Social Sciences
2004	BSc Combined Music and CSc (Major), Fac. of Fine Arts
2004	BSc Combined Geography and CSc (Major), Fac. of Social Sciences

E.2 Global Issues

Many of our Combined Programs were established in the traditional areas that appear to be a natural fit for computer science (Mathematics and Statistics). The newer programs attempt to bring together disciplines that are significantly different (e.g., Music and Computer Science, Visual Arts and Computer Science). These programs are very innovative and consequently there is little history or experience, in either our department, Canada or the rest of the world, on the best way to create and manage such interdisciplinary programs. In this appendix, we describe some of the challenges of these programs.

E.2.1 NUMBER OF PROGRAMS

Today our department offers a large number of different programs (i.e., major and minors in CSc, three options, eight combined programs, plus the software engineering program and its option). The resulting complexity raises at least a couple of issues.

- *Administrative:* Each program has its own set of requirements and regulations. The advisors must give guidance for all and has to be aware of all this information.

- *Recruiting:* During recruiting sessions there is not enough time to delve into the details of each program, and questions must be redirected and handled in a one-on-one fashion.

E.2.2 ENROLLMENT

At this point it is very difficult to identify precisely all the students enrolled in any or all combined programs (until they identify themselves), since these programs are resident in other faculties. It is important for the department to ensure it has accurate numbers for these programs in order to be able to plan effectively for their expansion.

E.2.3 CURRICULUM

There are courses specifically designed for students in a particular combined program. More, however, are needed in the future, if the combined programs are to become truly interdisciplinary. Within our own department, courses in fourth-year topics (e.g., in bioinformatics, music retrieval) are addressing this need.

Another challenge is typical of many interdisciplinary efforts, in that each of the two home departments for a program usually develops a list of required courses it believes its graduates should complete. One possible result is a program with a large list of required courses, and consequently few electives. While the impulse behind creating such course lists is understandable and can be justified academically, it is still the case that the students perceive the programs as very onerous. Consequently, the curricula of all of these programs are currently under revision.

E.3 The Geomatics Program

The Computer Science and Geography combined degree has been designed to develop geomatic professionals. There is a great deal of interest and need for geomatic specialists in industry, government, and academia. From the students' perspective this program offers them co-op job choices from two of the best co-op programs on campus: Computer Science and Geography. There is also much “hands-on” advising and mentoring available for students pursuing this program, from both Geography and Computer Science. The program has state-of-the-art labs and software, good instructors, and a very positive environment where all are encouraged to succeed.

From an administrative perspective, there exists good lines of communication between the Geomatics program in Geography and our department. The Geography Department is quick to respond to questions, and very keen to deepen the relationship with Computer Science to the mutual benefit of both programs. This communication at present is based around personal relationships between members in the two departments. This has led to a good start to the program, although we would also like to establish administrative links which can survive changes in personnel (e.g., study leave of faculty, changes in administrative officers, etc.).

From the student perspective, however, a major challenge exists around curriculum issues. A student must decide very early in their studies at UVic if they want to enroll in this program, as the consequences of starting late (i.e., in their second year) could make it very difficult for the student to complete their degree within four years. Geomatics is not particularly heavy in computer science, but it is critical for students to take CSC 110 and 115 in the first and second terms of their first year.

This is a practical concern as most incoming students do not understand the meaning of "Geomatics". Therefore they might need a year of more at university before discovering its meaning and whether or not they may find it an interesting study choice. Ways of addressing this include outreach to high schools and also outreach to first-year geography and computer science students.

E.4 The Music and Computer Science Program

The program is now in its third year and has undergone several calendar adjustments, with these changes primarily affecting the order in which courses are taken. It is also financed with base-budget funds. We soon expect graduates.

Students attracted to this program tend to have a strong music background, although these same students usually have weaker math and computer science backgrounds. One result is that these students find our fits courses, CSC 110 & CSC 115 challenging. We have attempted to address this in part by ensuring combined-degree students are scheduled in to the same first-year laboratory sessions, hence introducing them to their own cohort and also permitting laboratory instructors to target some aspects of in-lab exercises to the students' interest.

As with other combined programs, students in this program do find it a challenge to identify their academic home at UVic. However, Music & CSc students find it particularly challenging given they are trailblazers, that they have often found the program independent of each other and therefore do not know each other, and because there are not course outside the fourth-year level targeting specifically to the program. The School of Music has expressly invited the students to consider themselves part of the student community in that unit, and other attempts to form an esprit de corp have been social and academic meetings directly targeted towards these students.

In order to facilitate admission into the program, the application process is broken into two part. Students indicate their intention to apply to the program and are accepted as "pre-CSc/Music" students for that year, during which they are expected to enroll in the music courses required in the first year. (This first step is needed as the music program has limited resources and therefore places strict limits on the number of students who may enroll in its courses). After successfully completing this first year, students are accepted into the second year. This process, however, is confusing to the students, although a recent change has resulted in the administration of this two-step process being overseen by our department's Advising Officer. Another consequence of this process is that students must decide before their first year whether or not they wish to take this program; without doing so, they risk requiring more than four years to complete their major.

Appendix F: Software Engineering

There are two very different programs in software engineering offered by Computer Science. The first, the BSENG program, is offered jointly with the department of electrical and computer engineering, and the second is the software engineering option in the B. Sc. Computer Science degree.

F.1 The BSENG degree

The Bachelor of Software Engineering (BSENG) program has been designed to provide a solid and industrially relevant engineering education with focus on software engineering. The aim is to provide students with an education of lasting value that will maximize their potential to compete with the best software engineers in the world. The objective is to provide the students with a broad education in software engineering that will enable them to apply the principles of mathematics, science and engineering to identify, formulate and solve engineering problems.

The BSENG program has been designed to prepare students for a successful career in the software industry and the information technology sector by teaching them a breadth and depth of knowledge of software engineering that combines theory and best practices of computer science and engineering. BSENG students also acquire communication and business skills and become proficient in engineering design, particularly as it applies to software development and software systems.

The BSENG program aims to produce graduates who have the knowledge and ability to design and conduct experiments, to analyze and interpret data, to analyze and develop alternative solutions to problems, and to design systems, components, or processes to satisfy defined needs and criteria. It is designed to provide its graduates with effective communication skills, a thorough understanding of professional and ethical responsibilities, and the ability to work in a multi-disciplinary environment. The program is meant to instill in students the ability to self-learn, and a conviction that university education is only one stage in a life-long learning process. Creativity, resourcefulness, and active team participation and cooperation are promoted. The students are introduced to new and emerging technologies by faculty who are active in engineering and computer science research and development.

Through mandatory co-op, students have the opportunity to participate in one of the largest experiential learning programs in the country. This work experience is critical to providing students the opportunity to practice and apply knowledge and skills acquired throughout their studies in industrial settings and to sample different job prospects. Graduates of the BSENG program are prepared for the transition from student to responsible professional engineer, who are able to assess the impact of their work critically and recognize their obligations towards society.

F.1.1 THE BSENG PROGRAM

The BSENG program includes eight academic terms with 16-months of co-op terms intermixed.

The first year is common to all engineering programs at the University of Victoria and features an introduction to the role of software in the context of physical systems and real-world applications and emphasizes fundamental skills, including communication, programming, engineering design, and basic science.

The second and third year courses constitute the core of the program to develop software engineering, computer science and engineering knowledge and skills.

The fourth year provides an opportunity to specialize with technical electives and culminates with a significant design experience project based on the knowledge and skills acquired throughout the program. The final course in the BSENG program covers the legal, social and professional issues that arise in software engineering practice.

The knowledge and skills students acquire during the BSENG program include:

- an understanding of all aspects of software development and the software development process from the early design stages to long-term software maintenance and evolution;
- the ability to construct and evaluate software in the context of physical systems and real-world applications;
- the ability to apply engineering design principles to software development including trade-off analyses;
- an understanding of software quality criteria and assurance and the ability to assess the quality of a software system;
- the ability to plan and manage large software projects;
- the ability to work independently and collaboratively;
- an understanding of engineering economics and entrepreneurship in software practice;
- the ability to understand the underlying principles on which physical systems and real-world applications are built on;
- the ability to integrate and participate in the design process of these systems and applications;
- the capability to communicate effectively both orally and in writing; and
- a breadth of knowledge and skills in software engineering, as well as related areas of engineering, computer science, mathematics and complementary studies, that will provide a base for life-long learning.

The BSENG program is designed to house 75 students per year, for a total of 300 students over a 4 year program. The program began in September 2003 with First and Second Year and accepted 10 students into Second Year at that time. A couple of students joined the program and currently there are twelve students in fourth year. Approximately 70 students are currently in the entire BSENG program including an estimated 20-25 students in the Common First Year of Engineering.

F.1.2 ACCREDITATION FOR BSENG

The BSENG program is designed to be accredited by both the Canadian Engineering Accreditation Board (CEAB) and the Computer Science Accreditation Council (CSAC). The CEAB accreditation is currently ongoing. The Faculty went through the CEAB site visit in February 2007. Assuming the outcome of this process is successful, the Faculty will start the CSAC accreditation process. As an ongoing process, accreditation helps ensure that the BSENG program is kept up to date and effective and that the curriculum meets the guidelines of the Association of Professional Engineers and Geoscientists of British Columbia and CIPS.

Graduation from a nationally accredited program will enable students to become registered Professional Engineers (P.Eng) and obtain the ISP designation from CIPS.

F.1.3 THE BSENG CURRICULUM

The curriculum has been designed to give students a thorough understanding of the specification, development, implementation, testing, maintenance, and evolution of software systems ranging from embedded systems to large information systems. The courses ensure that students obtain a foundation in mathematics and basic sciences, a broad preparation in engineering design and engineering sciences, and an exposure to non-technical subjects in the form of complementary studies courses.

The requirements for the BSENG program include 47 courses and mandatory work experience of 16 months. The courses, designed to be taken over eight terms, include 38 prescribed courses and nine elective courses. The table below summarizes the components and the contributions of the key disciplines to the proposed software engineering degree program.

BSENG COMPONENT	NUMBER OF COURSES
Software Engineering	12
Engineering	7
Computer Science	6
Mathematics	6
Basic Sciences	4
Complementary Studies Electives	7
Engineering Electives	5
Total	47

The next table shows a suggested schedule for interleaving BSENG work terms and course terms. This schedule, if followed, allows students to graduate in a fixed amount of time (i.e., 4 2/3 years). Assuming that selected courses are offered multiple times, students have significant flexibility with respect to the length of their work terms.

Fall	Spring	Summer
Course Term 1A	Course Term 1B	Work Term or Off Term
Course Term 2A	Work Term or Off Term	Course Term 2B
Work Term or Off Term	Course Term 3A	Work Term or Off Term
Course Term 3B	Work Term or Off Term	Course Term 4A
Work Term or Off Term	Course Term 4B	Work Term or Off Term

F.2 The Software Engineering Option in Computer Science

Software Engineering is one of four options that students, with a Computer Science major, can choose to enhance their studies. The other three options are *Business, Mechatronics and Embedded Systems,* and *Bioinformatics*. The Computer Science Program with Software Engineering Option was established in 1998. It is intended for students of Computer Science who wish to specialize on applied concepts, methods and processes for systematically developing and maintaining software systems. It prepares students for careers in areas such as electronic commerce, medical software and systems, transportation systems and financial software. Participation in co-operative education is possible, but not mandatory, in the software engineering option.

F.2.1 SENG OPTION CURRICULUM

The curriculum for the SENG option was designed to give students a lot of flexibility to specialize in particular aspects of software engineering while still providing them with a sufficiently broad coverage of software engineering fundamentals. It also focuses on developing and improving communication skills, which are of key importance for software engineers. The UVic Calendar entry for this program can be found in Appendix J.

First Year. The first year concentrates on basic computer science and mathematics. It does not contain any specific software engineering courses. This is in contrast to other options that require fundamental knowledge about other disciplines, e.g., Bioinformatics (Biology) and Business (Economics).

Second Year. The second year contains SENG 265 (Software Development Methods) as a cornerstone mandatory software engineering course. It also prescribes courses on technical writing and statistics.

Third Year. Beyond the core computer science courses (databases, operating systems, programming languages, architecture and theory), the third year requires students to choose three software engineering courses from the following list of four topics: Human-Computer Interaction, Requirements and Formal Specifications, Object-Oriented Software Development, and Software Evolution.

Fourth Year. Year four requires students to take a course on Social and Professional Issues (ethics). Furthermore, students are required to take at least four more SENG courses at the 400 level. The current departmental calendar contains 18 courses at the 400 level (including the above-mentioned ethics course). An average of eight to ten courses on this level are typically being offered within the department in each given term.

Appendix G: Laboratories for CSC and SENG courses

Course	Title	Lab hours per week
CSC 100	Elementary Computing	2
CSC 105	Computers and Information Processing	2
CSC 110	Fundamentals of Programming: I	1
CSC 115	Fundamentals of Programming: II	1
CSC 160	Fundamentals of Programming: II for Engineers	1
CSC 212	The Practice of Computer Science	1
CSC 225	Algorithms and Data Structures: I	1
CSC 230	Algorithms and Data Structures: I	1.5
CSC 242	Computers in Science	2
CSC 355	Digital Logic and Computer Organization	2
CSC 360	Introduction to Operating Systems	1
CSC 375	Introduction to Systems Analysis	1
CSC 435	Compiler Construction	1
CSC 450	Computer Communications and Networks	3
CSC 460	Design and Analysis of Real-time Systems	3
CSC 461	Multimedia Systems	3
CSC 462	Distributed Computing	3
SENG 130	Engineering Software Systems	1
SENG 265	Software Development Methods	1
SENG 271	Software Architecture and Systems	1
SENG 321	Requirements Engineering and Formal Specifications	1
SENG 360	Security Engineering	1
SENG 371	Software Evolution	1
SENG 410	Media Applications	3
SENG 466	Software for Embedded and Mechatronics Systems	1
SENG 474	Data Mining	1

Appendix H: Two Proposals for New Options

One result of the department's recent growth is that we now have three regular faculty engaged in Computer Graphics research and four regular faculty engaged in Computer Networks research. Given the attractiveness of these two topics to students (i.e., our one fourth-year Graphics course and one fourth-year Networking course are nearly always full), we are investigating two new options. Discussions for the Networking Option have proceeded as far as the department's Undergraduate Curriculum Committee, which has approved the program for the consideration of the whole department. Our Graphics option proposal is still under discussion, and we expect the result to be offered to the Curriculum committee for review sometime in the next four months.

H.1 Networking Option

The proposal for a "Computer Communications & Networks Option" consists of:

- Creation of a new third-year course, CSC 361, which will serve as the first networking course in the program.
- Deletion of our existing fourth-year networking course, CSC 450.
- Addition of three other fourth-year courses which reflect the specialities of the department's research group. All of these courses were offered previously as special-topics courses.
- Changes to the prerequisites of several other Computer Science courses and the description of two other programs.

NEW THIRD-YEAR COURSE: CSC 361

This course, entitled "Computer Communications and Networks", is intended to replace the existing fourth-year course (CSC 450) which will be phased out. The material will be similar to that of the old course, but at a level suitable for students in third year and with prerequisites from the second-year level (CSC 225, 230 and SENG 265). It is also the foundation course for each of a set of proposed fourth-year courses. This course sequencing is more appropriate than the present arrangement, as currently the fourth-year introductory course is a prerequisite for any fourth-year special topics course devoted to networking. Students will be better able to schedule their programs of study if they wish to pursue an interest in networking.

A weekly three-hour lab is also part of this course.

NEW FOURTH-YEAR COURSE: CSC 463 (WIRELESS AND MOBILE NETWORKS)

This is an introduction to selected issues in wireless and mobile networks. Topics include radio basics, mobility models, location management, QoS (Quality of Service), MAC (Medium Access Control), routing, and transport protocols over different types of wireless and mobile networks. The one prerequisite is the new CSC 361 or its equivalent. There are no labs scheduled for this course.

NEW FOURTH-YEAR COURSE: CSC 465 (SWITCHING, TRAFFIC AND QOS)

This course emphasizes concepts involving layers 2 and 3 of the traditional OSI networking model. It begins with a review of computer communications and networks, and then proceeds to switch & router architectures, followed by traffic management & traffic engineering. Other topics include data and control plane technologies (e.g., virtual private networks, multi-protocol label switching, etc.), QoS routing aspects, other protocols, and optical networks. The one prerequisite is the new CSC 361 or its equivalent. There are no labs for this course.

NEW FOURTH-YEAR COURSE: CSC 466 (OVERLAY AND PEER-TO-PEER NETWORKING)

This course focuses on layer 3 and above of the traditional OSI networking model, and includes the control plane of the Internet. Topics include: overlay network architectures, peer-to-peer application models, end-to-end control mechanisms, inter- and intra-domain routing protocols, servicing provisioning, network measurement, and related best practices on the Internet. The one prerequisite is the new CSC 361 or its equivalent. There are no labs for this course.

H.1.1 IMPACT ON EXISTING COURSES AND PROGRAMS

Several courses in our program which are regularly offered already have CSC 450 as a prerequisite. These are CSC 461 (“Multimedia Systems”) and CSC 462 (“Distributed Systems”). Using CSC 361 instead as a prerequisite is suitable for these courses.

The Bachelor of Software Engineering (BSENG) program currently includes CSC 450 as a course in its Term 4A. After consulting with the BSENG program director, our recommendation for that program is CSC 361 be taken in Term 4A.

A similar situation exists with the Combined Major in Music and Computer Science as CSC 450 is listed as an elective in the fourth year. Again, or recommendation is similar, and that the reference to CSC 450 be replaced with CSC 361.

One last program affected is our own Major and Honours degrees with the “Systems” emphasis. Here we may simply replace CSC 450 by CSC 361.

H.2 Graphics Options

The only graphics course listed in the 2006-07 UVic Calendar is CSC 405, *Introduction to Computer Graphics*. This course has been deleted from the upcoming new calendar and has been replaced by a new third year introductory graphics course, CSC 305.

CSC 305 Introduction to Computer Graphics. *Prerequisites:* CSC 225, MATH 100 and either MATH 133 or 233A.
Introduction to computer graphics. Principles of raster image generation. Example of a graphics API. Graphics primitives, data structures. Coordinate systems, affine transformations and viewing of graphical objects. Introduction to rendering including shading models and ray tracing. Introduction to modeling including polygon meshes, subdivision, and parametric curves and surfaces, colour.

All the proposed new 4th year courses could be taught as joint undergraduate/graduate courses by giving the graduate students additional work in the form of projects.

CSC 4XX/5XX Fundamentals of Computer Animation. *Prerequisites:* CSC 305.
Principles of traditional animation, key framing, parametric and track animation, free form deformation, inverse kinematics, dynamics, spring mass systems, particle systems, numerical integration, Lagrangian constraints, space time constraints, collisions, human animation, behavioural animation, metamorphosis, implicit animation techniques, animating liquids, gases and cloth, motion capture. Additionally, we will discuss animation interfaces (such as Maya) and introduce MEL scripting.

CSC 4XX/5XX Fundamentals of Computer Modeling. *Prerequisites:* CSC 305.
Parametric Modeling. B-splines and NURBS. Subdivision schemes. Surface subdivision. Multiresolution. Wavelets. Implicit modeling. Blends. Polygonization. Blobtree. Precise contact modeling. Solid modeling. CSG. Procedural modeling, sketch based modeling / meshing / point based modeling.

CSC 4XX/5XX Fundamentals of Computer Rendering. Prerequisites: CSC 305.
Physical foundations of illuminations techniques. Colour. Radiometry and photometry. Reflection models. The rendering equation. Ray tracing. Monte Carlo techniques. Sampling and antialiasing. Texturing. Radiosity. Photon tracing. Volume rendering. Image-based rendering. Real-time shading, NPR, Image based rendering, point based rendering.

The following are some proposed Special Topics courses.

CSC 4XX Introduction to Computer Game Production and Development. *Prerequisites:* CSC 225, MATH 133 or 233A.
This course will cover 2D and 3D graphics, as well as topics such as performance sensitivity, reliability and compelling game play. Course projects will involve developing, debugging and optimizing games for multiple hardware platforms including, cell phones, laptops, PDAs such as the Pocket PC, as well as desktop PCs. The class will be taught using modern languages (C++), operating systems, and development tools. Students will review current trends in computer game programming and build their own 2D and 3D games on top of available game engines.

CSC 4XX Tools and Technology for the WWW (Non-computer scientists). *Prerequisites:* None.
Introduction to the theory and practice of developing sites on and technology for the World Wide Web. The course covers HTML, JavaScript, ASP, and CGI programming.

CSC 4XX Introduction to Computer Graphics Animation (or Animation for non-computer scientists). *Prerequisites:* None.
This introductory course allows students to explore computer animation through the use of the Maya software animation package. During the course, students develop skills in digital modeling, basic texturing and simple rendering in order to understand the steps and procedures that lead to the completion of animations. A key element of this course is the design relationship between virtual and actual objects, such the design of computer graphics models based on abstracting elements of the real world. The class is project based with the final goal of students being able to create simple computer generated animations. No prerequisites; no programming; No technical background necessary; Artists and Non-Artists are encouraged to enroll.

CSC 4XX/5XX Non-Photorealistic Rendering (Advanced Undergraduate and Graduate) . *Prerequisites*: CSC 305.
In many applications, a non-photorealistic (NPR) image has advantages over a photorealistic image. NPR images omit extraneous detail, focus attention on relevant features, clarify, simplify, and disambiguate shape, and show hidden parts. This course will cover current work in the area of NPR and will give students an opportunity to work on an NPR application.

Non-Photorealistic Rendering is an advanced seminar course in computer graphics. It serves as an introduction to advanced topics and research in the field. The course is open to graduate students and advanced undergraduates. The prerequisite for the course is CSC 405 or instructor approval. In this seminar course, students will read, present, and discuss advanced graphics techniques. In addition, students will also implement one of these techniques in a programming project. The NPR project can be implemented in a programming language of the students’ choice (C++, Java, OpenGL, etc.). Students may implement the project individually or with the instructor's approval, in groups.

CSC 4XX/5XX Perception and Graphics (Advanced Undergraduate and Graduate). *Prerequisites*: CSC 305.
This course provides students with: a general background knowledge of the human visual system (HVS), ethical and legal instruction for studies involving human subjects, Institutional Review Board (IRB) policies and procedures, how knowledge of the HVS can be used to create more effective computer graphics imagery and visualizations, and the task based evaluation of computer graphics imagery. In addition, the course will cover current graphics research that seeks to facilitate a viewer in understanding and interpreting computer image as well as research conducted by members of the vision sciences community who are using computer graphics to facilitate the investigation of fundamental processes of perception.

Perception and Graphics is an advanced seminar course in computer graphics. It serves as an introduction to advanced topics and research in the field. The course is open to graduate students and advanced undergraduates. The prerequisite for the course is CSC 405 or instructor approval. In this seminar course, students will read, present, and discuss advanced graphics techniques. In addition, students will also implement one of these techniques in a programming project and evaluate the resulting computer imagery in a task based study. The project can be implemented in a programming language of the students' choice (C++, Java, OpenGL, etc.). Students may implement the project individually or, with the instructor's approval, in groups

CSC 5XX Implicit Modeling (Graduate). *Prerequisites:* CSC 305.
A detailed look at modeling using implicit and iso-surface techniques taking an in depth review of the literature. Point sets, radial basis functions, distance surfaces, level sets. Algebraic methods, skeletal models, field function design, modeling techniques, rendering and texture mapping. Polygonization algorithms, ray tracing implicits, techniques for animation, metamorphosis, precise contact modeling, deformation and warping. Algorithms and data structures and implementation details will be presented. Students will be required to implement a relevant programming project.

Finally, in addition to the above, some potential future specialized Topics courses could be offered in Computational Aesthetics, Image processing / Computer Vision and Computational Photography.

H.3 Ensuring Success of Newly-Introduced Options

The department recognizes that the first step in creating a new program is submitting a Calendar Entry to the Faculty and then to Senate. However, other elements are required for the success of such a program. These new programs require considerable attention for the first three years of their existence. Some examples of activities are:

- Definition and regular revision of learning inputs and outcomes.
- Ongoing revision and development of curriculum.
- Assessment of suitability for the target audience, using interviews and enrolment monitoring (as two examples).
- Promotion with the target audience.
- Liasing with other Programs and Curriculum committees.

The scope of such attention is larger than that of traditional Course Champions (as is our department's suggested practice for lower-level courses.) Consequently, we are formulating a proposal to create the role of Program Champion for the CC&N program, and once a Graphics proposal has been considered, a similar Program Champion for the Graphics option. This role may be performed as part of the administrative duties of a member of regular faculty, or if this is considered infeasible by the Department Chair, as a task contracted out to someone with experience developing and delivering curriculum (e.g., a professor emeritus, adjunct professor, etc.).

Appendix I: Service Courses — Observations and Challenges

I.1 History

- CSC 100 has been taught since 1979, and it has had well over 10,000 students over the past 20 years.
- CSC 105 was first taught in 1992, with approximately 6,000 students over the past 15 years
- CSC 110 was first taught in 1979, and it has had well over 10,000 students over the past 20 years.
- CSC 115 was first taught in 1979, between 5,000 and 10,000 students over the past 20 years.
- CSC160 was first taught in the 1985, and is a CSC 115 course designed specifically for Engineering students. 2007 is the final year in which it is being offered.
- CSC111 is a new course to be offered for the first time in Fall 2007. It is designed specifically for Engineering students.

I.2 Observations

These courses have had thousands of students in them over the past five years. Over 90% of these students do not take another CSc course. We should have, as a goal, the recruitment of 20% of these students into at least one other CSc course.

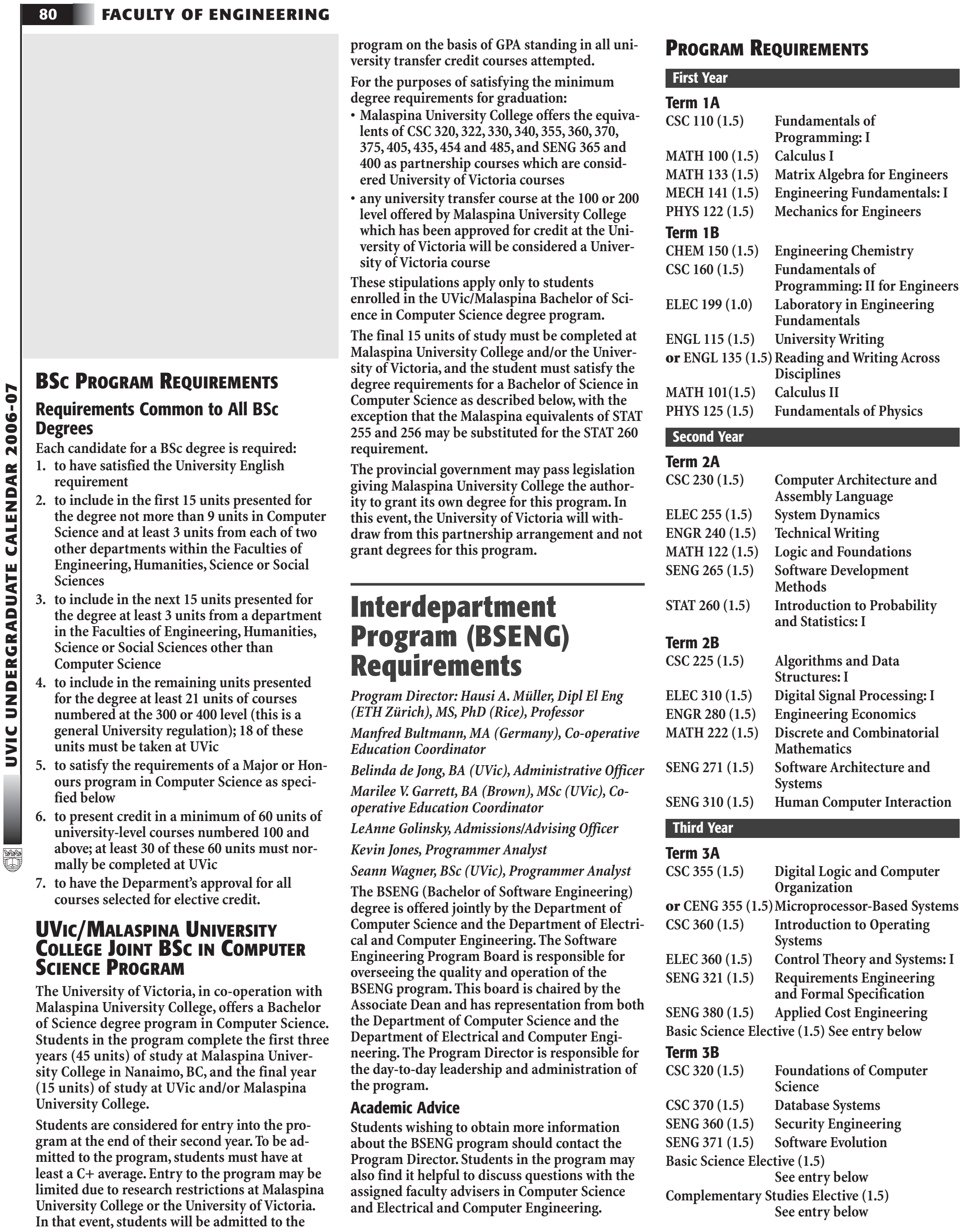
Currently the only course that regularly sees some students from the Faculty of Education is CSC 100. We believe that we could team with Education to offer a service course of *Computer Science in Education,* at the 2nd year level. We do have the expertise here, and its effectiveness could be increased if we teamed with a faculty member from Education. Another area to explore is having 2nd year service courses, such as "Website Design and Creation" which could be well received.

Service courses are valuable on a number of fronts:

- These courses expose students to computer science thought and methods. It gives students a peek into how things are done in our discipline.
- Our service course students take away a whole suite of new skills. More importantly, students gain insights into things that they never thought they could even begin to understand.
- Our service course students take away skills that are excellent building blocks for co-op students — no matter which co-op program they enter.
- Getting non-CSc students into our buildings, labs, our Course Union, our Women in Engineering and Computer Science, any of the areas/groups where they can socialize, is good for both us (Computer Scientists) and them. It ensures that we as computer scientists look outward, and it is a good opportunity for our service-course students to see that computer scientists do not confirm to traditional media stereotypes of IT professionals.

Appendix J: Calendar Descriptions of Programs

J.1 UVic/Malaspina University College Joint BSc; BSENG Program



80 FACULTY OF ENGINEERING

UVIC UNDERGRADUATE CALENDAR 2006-07

BSc Program Requirements

Requirements Common to All BSc Degrees

Each candidate for a BSc degree is required:

1. to have satisfied the University English requirement
2. to include in the first 15 units presented for the degree not more than 9 units in Computer Science and at least 3 units from each of two other departments within the Faculties of Engineering, Humanities, Science or Social Sciences
3. to include in the next 15 units presented for the degree at least 3 units from a department in the Faculties of Engineering, Humanities, Science or Social Sciences other than Computer Science
4. to include in the remaining units presented for the degree at least 21 units of courses numbered at the 300 or 400 level (this is a general University regulation); 18 of these units must be taken at UVic
5. to satisfy the requirements of a Major or Honours program in Computer Science as specified below
6. to present credit in a minimum of 60 units of university-level courses numbered 100 and above; at least 30 of these 60 units must normally be completed at UVic
7. to have the Deparment's approval for all courses selected for elective credit.

UVic/Malaspina University College Joint BSc in Computer Science Program

The University of Victoria, in co-operation with Malaspina University College, offers a Bachelor of Science degree program in Computer Science. Students in the program complete the first three years (45 units) of study at Malaspina University College in Nanaimo, BC, and the final year (15 units) of study at UVic and/or Malaspina University College.

Students are considered for entry into the program at the end of their second year. To be admitted to the program, students must have at least a C+ average. Entry to the program may be limited due to research restrictions at Malaspina University College or the University of Victoria. In that event, students will be admitted to the program on the basis of GPA standing in all university transfer credit courses attempted.

For the purposes of satisfying the minimum degree requirements for graduation:

- Malaspina University College offers the equivalents of CSC 320, 322, 330, 340, 355, 360, 370, 375, 405, 435, 454 and 485, and SENG 365 and 400 as partnership courses which are considered University of Victoria courses
- any university transfer course at the 100 or 200 level offered by Malaspina University College which has been approved for credit at the University of Victoria will be considered a University of Victoria course

These stipulations apply only to students enrolled in the UVic/Malaspina Bachelor of Science in Computer Science degree program.

The final 15 units of study must be completed at Malaspina University College and/or the University of Victoria, and the student must satisfy the degree requirements for a Bachelor of Science in Computer Science as described below, with the exception that the Malaspina equivalents of STAT 255 and 256 may be substituted for the STAT 260 requirement.

The provincial government may pass legislation giving Malaspina University College the authority to grant its own degree for this program. In this event, the University of Victoria will withdraw from this partnership arrangement and not grant degrees for this program.

Interdepartment Program (BSENG) Requirements

Program Director: Hausi A. Müller, Dipl El Eng (ETH Zürich), MS, PhD (Rice), Professor

Manfred Bultmann, MA (Germany), Co-operative Education Coordinator

Belinda de Jong, BA (UVic), Administrative Officer

Marilee V. Garrett, BA (Brown), MSc (UVic), Co-operative Education Coordinator

LeAnne Golinsky, Admissions/Advising Officer

Kevin Jones, Programmer Analyst

Seann Wagner, BSc (UVic), Programmer Analyst

The BSENG (Bachelor of Software Engineering) degree is offered jointly by the Department of Computer Science and the Department of Electrical and Computer Engineering. The Software Engineering Program Board is responsible for overseeing the quality and operation of the BSENG program. This board is chaired by the Associate Dean and has representation from both the Department of Computer Science and the Department of Electrical and Computer Engineering. The Program Director is responsible for the day-to-day leadership and administration of the program.

Academic Advice

Students wishing to obtain more information about the BSENG program should contact the Program Director. Students in the program may also find it helpful to discuss questions with the assigned faculty advisers in Computer Science and Electrical and Computer Engineering.

Program Requirements

First Year

Term 1A

CSC 110 (1.5)	Fundamentals of Programming: I
MATH 100 (1.5)	Calculus I
MATH 133 (1.5)	Matrix Algebra for Engineers
MECH 141 (1.5)	Engineering Fundamentals: I
PHYS 122 (1.5)	Mechanics for Engineers

Term 1B

CHEM 150 (1.5)	Engineering Chemistry
CSC 160 (1.5)	Fundamentals of Programming: II for Engineers
ELEC 199 (1.0)	Laboratory in Engineering Fundamentals
ENGL 115 (1.5)	University Writing
or ENGL 135 (1.5)	Reading and Writing Across Disciplines
MATH 101(1.5)	Calculus II
PHYS 125 (1.5)	Fundamentals of Physics

Second Year

Term 2A

CSC 230 (1.5)	Computer Architecture and Assembly Language
ELEC 255 (1.5)	System Dynamics
ENGR 240 (1.5)	Technical Writing
MATH 122 (1.5)	Logic and Foundations
SENG 265 (1.5)	Software Development Methods
STAT 260 (1.5)	Introduction to Probability and Statistics: I

Term 2B

CSC 225 (1.5)	Algorithms and Data Structures: I
ELEC 310 (1.5)	Digital Signal Processing: I
ENGR 280 (1.5)	Engineering Economics
MATH 222 (1.5)	Discrete and Combinatorial Mathematics
SENG 271 (1.5)	Software Architecture and Systems
SENG 310 (1.5)	Human Computer Interaction

Third Year

Term 3A

CSC 355 (1.5)	Digital Logic and Computer Organization
or CENG 355 (1.5)	Microprocessor-Based Systems
CSC 360 (1.5)	Introduction to Operating Systems
ELEC 360 (1.5)	Control Theory and Systems: I
SENG 321 (1.5)	Requirements Engineering and Formal Specification
SENG 380 (1.5)	Applied Cost Engineering

Basic Science Elective (1.5) See entry below

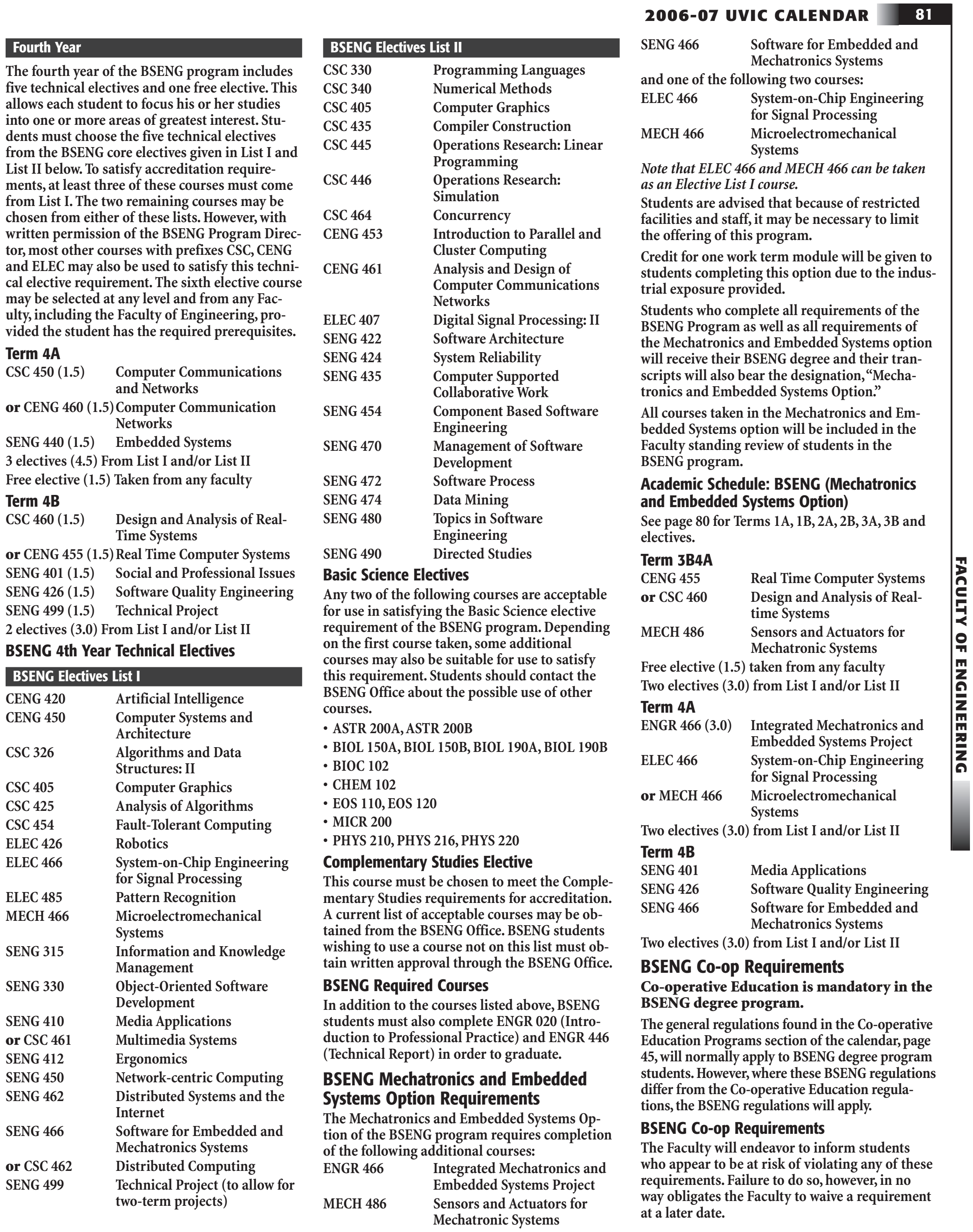
Term 3B

CSC 320 (1.5)	Foundations of Computer Science
CSC 370 (1.5)	Database Systems
SENG 360 (1.5)	Security Engineering
SENG 371 (1.5)	Software Evolution

Basic Science Elective (1.5) See entry below

Complementary Studies Elective (1.5) See entry below

BSENG Program, continued



Fourth Year

The fourth year of the BSENG program includes five technical electives and one free elective. This allows each student to focus his or her studies into one or more areas of greatest interest. Students must choose the five technical electives from the BSENG core electives given in List I and List II below. To satisfy accreditation requirements, at least three of these courses must come from List I. The two remaining courses may be chosen from either of these lists. However, with written permission of the BSENG Program Director, most other courses with prefixes CSC, CENG and ELEC may also be used to satisfy this technical elective requirement. The sixth elective course may be selected at any level and from any Faculty, including the Faculty of Engineering, provided the student has the required prerequisites.

Term 4A

CSC 450 (1.5)	Computer Communications and Networks
or CENG 460 (1.5)	Computer Communication Networks
SENG 440 (1.5)	Embedded Systems
3 electives (4.5)	From List I and/or List II
Free elective (1.5)	Taken from any faculty

Term 4B

CSC 460 (1.5)	Design and Analysis of Real-Time Systems
or CENG 455 (1.5)	Real Time Computer Systems
SENG 401 (1.5)	Social and Professional Issues
SENG 426 (1.5)	Software Quality Engineering
SENG 499 (1.5)	Technical Project
2 electives (3.0)	From List I and/or List II

BSENG 4th Year Technical Electives

BSENG Electives List I

CENG 420	Artificial Intelligence
CENG 450	Computer Systems and Architecture
CSC 326	Algorithms and Data Structures: II
CSC 405	Computer Graphics
CSC 425	Analysis of Algorithms
CSC 454	Fault-Tolerant Computing
ELEC 426	Robotics
ELEC 466	System-on-Chip Engineering for Signal Processing
ELEC 485	Pattern Recognition
MECH 466	Microelectromechanical Systems
SENG 315	Information and Knowledge Management
SENG 330	Object-Oriented Software Development
SENG 410	Media Applications
or CSC 461	Multimedia Systems
SENG 412	Ergonomics
SENG 450	Network-centric Computing
SENG 462	Distributed Systems and the Internet
SENG 466	Software for Embedded and Mechatronics Systems
or CSC 462	Distributed Computing
SENG 499	Technical Project (to allow for two-term projects)

BSENG Electives List II

CSC 330	Programming Languages
CSC 340	Numerical Methods
CSC 405	Computer Graphics
CSC 435	Compiler Construction
CSC 445	Operations Research: Linear Programming
CSC 446	Operations Research: Simulation
CSC 464	Concurrency
CENG 453	Introduction to Parallel and Cluster Computing
CENG 461	Analysis and Design of Computer Communications Networks
ELEC 407	Digital Signal Processing: II
SENG 422	Software Architecture
SENG 424	System Reliability
SENG 435	Computer Supported Collaborative Work
SENG 454	Component Based Software Engineering
SENG 470	Management of Software Development
SENG 472	Software Process
SENG 474	Data Mining
SENG 480	Topics in Software Engineering
SENG 490	Directed Studies

Basic Science Electives

Any two of the following courses are acceptable for use in satisfying the Basic Science elective requirement of the BSENG program. Depending on the first course taken, some additional courses may also be suitable for use to satisfy this requirement. Students should contact the BSENG Office about the possible use of other courses.

- ASTR 200A, ASTR 200B
- BIOL 150A, BIOL 150B, BIOL 190A, BIOL 190B
- BIOC 102
- CHEM 102
- EOS 110, EOS 120
- MICR 200
- PHYS 210, PHYS 216, PHYS 220

Complementary Studies Elective

This course must be chosen to meet the Complementary Studies requirements for accreditation. A current list of acceptable courses may be obtained from the BSENG Office. BSENG students wishing to use a course not on this list must obtain written approval through the BSENG Office.

BSENG Required Courses

In addition to the courses listed above, BSENG students must also complete ENGR 020 (Introduction to Professional Practice) and ENGR 446 (Technical Report) in order to graduate.

BSENG Mechatronics and Embedded Systems Option Requirements

The Mechatronics and Embedded Systems Option of the BSENG program requires completion of the following additional courses:

ENGR 466	Integrated Mechatronics and Embedded Systems Project
MECH 486	Sensors and Actuators for Mechatronic Systems
SENG 466	Software for Embedded and Mechatronics Systems

and one of the following two courses:

ELEC 466	System-on-Chip Engineering for Signal Processing
MECH 466	Microelectromechanical Systems

Note that ELEC 466 and MECH 466 can be taken as an Elective List I course.

Students are advised that because of restricted facilities and staff, it may be necessary to limit the offering of this program.

Credit for one work term module will be given to students completing this option due to the industrial exposure provided.

Students who complete all requirements of the BSENG Program as well as all requirements of the Mechatronics and Embedded Systems option will receive their BSENG degree and their transcripts will also bear the designation, "Mechatronics and Embedded Systems Option."

All courses taken in the Mechatronics and Embedded Systems option will be included in the Faculty standing review of students in the BSENG program.

Academic Schedule: BSENG (Mechatronics and Embedded Systems Option)

See page 80 for Terms 1A, 1B, 2A, 2B, 3A, 3B and electives.

Term 3B4A

CENG 455	Real Time Computer Systems
or CSC 460	Design and Analysis of Real-time Systems
MECH 486	Sensors and Actuators for Mechatronic Systems
Free elective (1.5)	taken from any faculty
Two electives (3.0)	from List I and/or List II

Term 4A

ENGR 466 (3.0)	Integrated Mechatronics and Embedded Systems Project
ELEC 466	System-on-Chip Engineering for Signal Processing
or MECH 466	Microelectromechanical Systems
Two electives (3.0)	from List I and/or List II

Term 4B

SENG 401	Media Applications
SENG 426	Software Quality Engineering
SENG 466	Software for Embedded and Mechatronics Systems
Two electives (3.0)	from List I and/or List II

BSENG Co-op Requirements

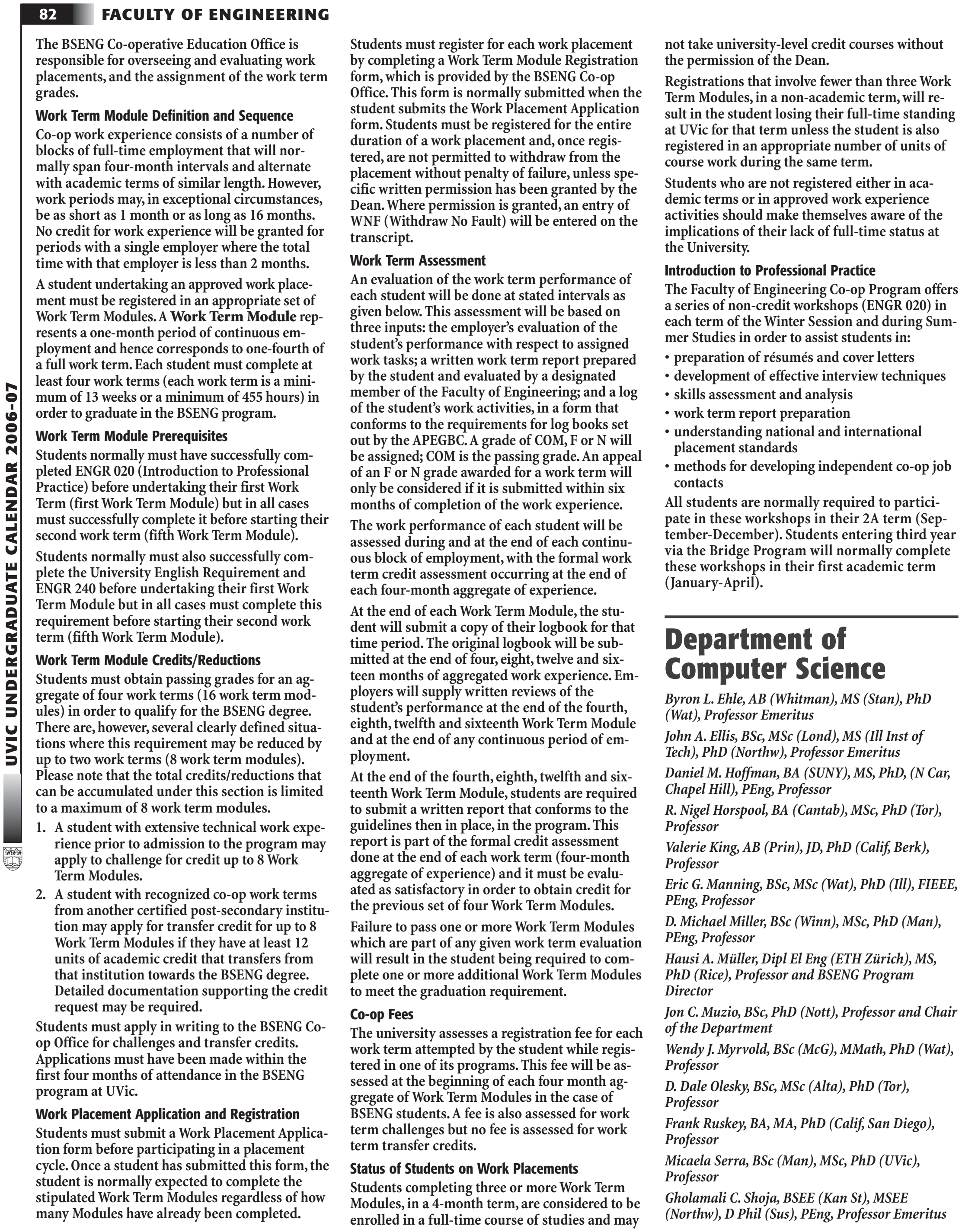
Co-operative Education is mandatory in the BSENG degree program.

The general regulations found in the Co-operative Education Programs section of the calendar, page 45, will normally apply to BSENG degree program students. However, where these BSENG regulations differ from the Co-operative Education regulations, the BSENG regulations will apply.

BSENG Co-op Requirements

The Faculty will endeavor to inform students who appear to be at risk of violating any of these requirements. Failure to do so, however, in no way obligates the Faculty to waive a requirement at a later date.

BSENG Program, continued; Computer Science Undergraduate Programs



The BSENG Co-operative Education Office is responsible for overseeing and evaluating work placements, and the assignment of the work term grades.

Work Term Module Definition and Sequence

Co-op work experience consists of a number of blocks of full-time employment that will normally span four-month intervals and alternate with academic terms of similar length. However, work periods may, in exceptional circumstances, be as short as 1 month or as long as 16 months. No credit for work experience will be granted for periods with a single employer where the total time with that employer is less than 2 months.

A student undertaking an approved work placement must be registered in an appropriate set of Work Term Modules. A **Work Term Module** represents a one-month period of continuous employment and hence corresponds to one-fourth of a full work term. Each student must complete at least four work terms (each work term is a minimum of 13 weeks or a minimum of 455 hours) in order to graduate in the BSENG program.

Work Term Module Prerequisites

Students normally must have successfully completed ENGR 020 (Introduction to Professional Practice) before undertaking their first Work Term (first Work Term Module) but in all cases must successfully complete it before starting their second work term (fifth Work Term Module).

Students normally must also successfully complete the University English Requirement and ENGR 240 before undertaking their first Work Term Module but in all cases must complete this requirement before starting their second work term (fifth Work Term Module).

Work Term Module Credits/Reductions

Students must obtain passing grades for an aggregate of four work terms (16 work term modules) in order to qualify for the BSENG degree. There are, however, several clearly defined situations where this requirement may be reduced by up to two work terms (8 work term modules). Please note that the total credits/reductions that can be accumulated under this section is limited to a maximum of 8 work term modules.

1. A student with extensive technical work experience prior to admission to the program may apply to challenge for credit up to 8 Work Term Modules.
2. A student with recognized co-op work terms from another certified post-secondary institution may apply for transfer credit for up to 8 Work Term Modules if they have at least 12 units of academic credit that transfers from that institution towards the BSENG degree. Detailed documentation supporting the credit request may be required.

Students must apply in writing to the BSENG Co-op Office for challenges and transfer credits. Applications must have been made within the first four months of attendance in the BSENG program at UVic.

Work Placement Application and Registration

Students must submit a Work Placement Application form before participating in a placement cycle. Once a student has submitted this form, the student is normally expected to complete the stipulated Work Term Modules regardless of how many Modules have already been completed.

Students must register for each work placement by completing a Work Term Module Registration form, which is provided by the BSENG Co-op Office. This form is normally submitted when the student submits the Work Placement Application form. Students must be registered for the entire duration of a work placement and, once registered, are not permitted to withdraw from the placement without penalty of failure, unless specific written permission has been granted by the Dean. Where permission is granted, an entry of WNF (Withdraw No Fault) will be entered on the transcript.

Work Term Assessment

An evaluation of the work term performance of each student will be done at stated intervals as given below. This assessment will be based on three inputs: the employer's evaluation of the student's performance with respect to assigned work tasks; a written work term report prepared by the student and evaluated by a designated member of the Faculty of Engineering; and a log of the student's work activities, in a form that conforms to the requirements for log books set out by the APEGBC. A grade of COM, F or N will be assigned; COM is the passing grade. An appeal of an F or N grade awarded for a work term will only be considered if it is submitted within six months of completion of the work experience.

The work performance of each student will be assessed during and at the end of each continuous block of employment, with the formal work term credit assessment occurring at the end of each four-month aggregate of experience.

At the end of each Work Term Module, the student will submit a copy of their logbook for that time period. The original logbook will be submitted at the end of four, eight, twelve and sixteen months of aggregated work experience. Employers will supply written reviews of the student's performance at the end of the fourth, eighth, twelfth and sixteenth Work Term Module and at the end of any continuous period of employment.

At the end of the fourth, eighth, twelfth and sixteenth Work Term Module, students are required to submit a written report that conforms to the guidelines then in place, in the program. This report is part of the formal credit assessment done at the end of each work term (four-month aggregate of experience) and it must be evaluated as satisfactory in order to obtain credit for the previous set of four Work Term Modules.

Failure to pass one or more Work Term Modules which are part of any given work term evaluation will result in the student being required to complete one or more additional Work Term Modules to meet the graduation requirement.

Co-op Fees

The university assesses a registration fee for each work term attempted by the student while registered in one of its programs. This fee will be assessed at the beginning of each four month aggregate of Work Term Modules in the case of BSENG students. A fee is also assessed for work term challenges but no fee is assessed for work term transfer credits.

Status of Students on Work Placements

Students completing three or more Work Term Modules, in a 4-month term, are considered to be enrolled in a full-time course of studies and may not take university-level credit courses without the permission of the Dean.

Registrations that involve fewer than three Work Term Modules, in a non-academic term, will result in the student losing their full-time standing at UVic for that term unless the student is also registered in an appropriate number of units of course work during the same term.

Students who are not registered either in academic terms or in approved work experience activities should make themselves aware of the implications of their lack of full-time status at the University.

Introduction to Professional Practice

The Faculty of Engineering Co-op Program offers a series of non-credit workshops (ENGR 020) in each term of the Winter Session and during Summer Studies in order to assist students in:

- preparation of résumés and cover letters
- development of effective interview techniques
- skills assessment and analysis
- work term report preparation
- understanding national and international placement standards
- methods for developing independent co-op job contacts

All students are normally required to participate in these workshops in their 2A term (September-December). Students entering third year via the Bridge Program will normally complete these workshops in their first academic term (January-April).

Department of Computer Science

Byron L. Ehle, AB (Whitman), MS (Stan), PhD (Wat), Professor Emeritus

John A. Ellis, BSc, MSc (Lond), MS (Ill Inst of Tech), PhD (Northw), Professor Emeritus

Daniel M. Hoffman, BA (SUNY), MS, PhD, (N Car, Chapel Hill), PEng, Professor

R. Nigel Horspool, BA (Cantab), MSc, PhD (Tor), Professor

Valerie King, AB (Prin), JD, PhD (Calif, Berk), Professor

Eric G. Manning, BSc, MSc (Wat), PhD (Ill), FIEEE, PEng, Professor

D. Michael Miller, BSc (Winn), MSc, PhD (Man), PEng, Professor

Hausi A. Müller, Dipl El Eng (ETH Zürich), MS, PhD (Rice), Professor and BSENG Program Director

Jon C. Muzio, BSc, PhD (Nott), Professor and Chair of the Department

Wendy J. Myrvold, BSc (McG), MMath, PhD (Wat), Professor

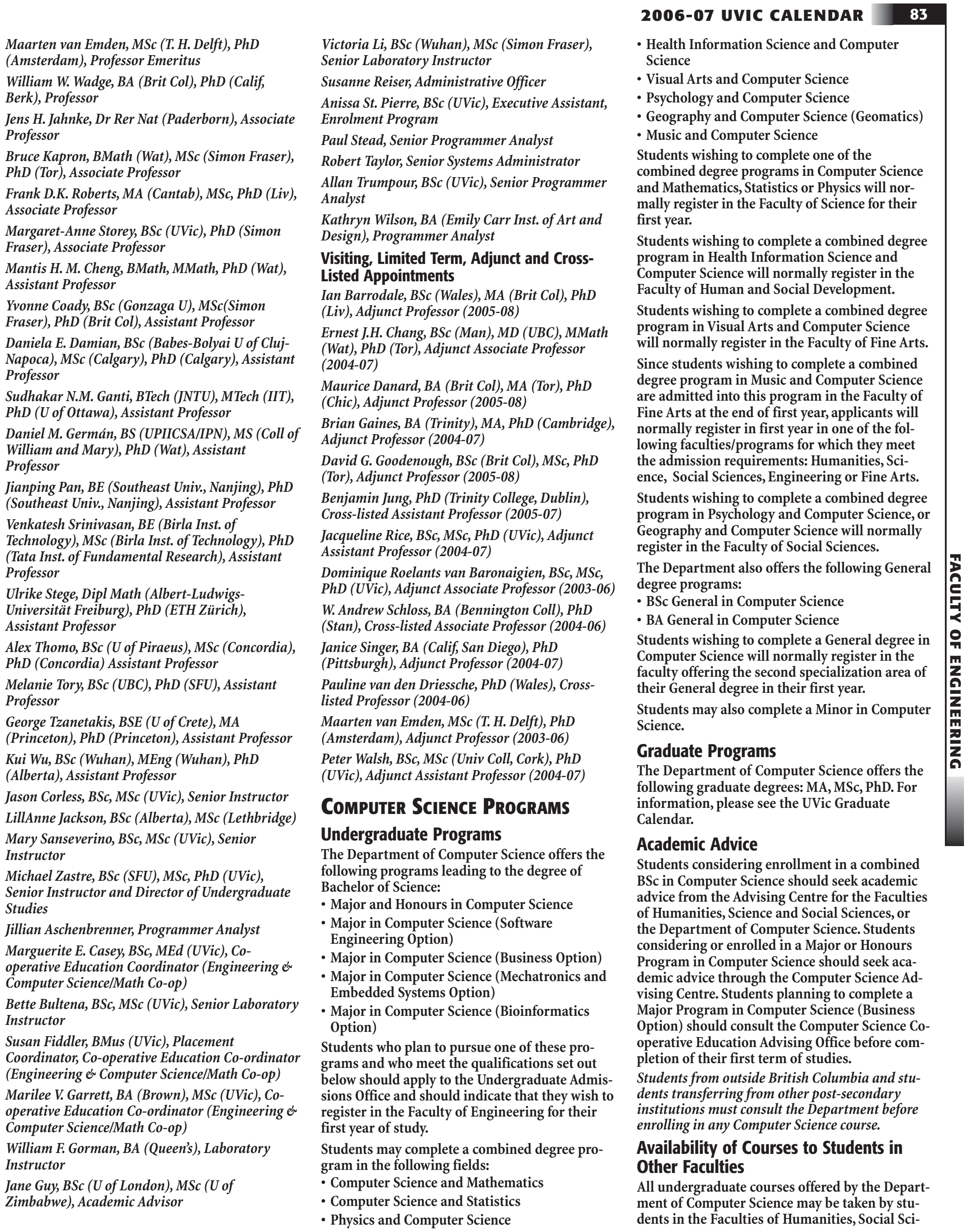
D. Dale Olesky, BSc, MSc (Alta), PhD (Tor), Professor

Frank Ruskey, BA, MA, PhD (Calif, San Diego), Professor

Micaela Serra, BSc (Man), MSc, PhD (UVic), Professor

Gholamali C. Shoja, BSEE (Kan St), MSEE (Northw), D Phil (Sus), PEng, Professor Emeritus

Computer Science Undergraduate Programs, continued



Maarten van Emden, MSc (T. H. Delft), PhD (Amsterdam), Professor Emeritus

William W. Wadge, BA (Brit Col), PhD (Calif, Berk), Professor

Jens H. Jahnke, Dr Rer Nat (Paderborn), Associate Professor

Bruce Kapron, BMath (Wat), MSc (Simon Fraser), PhD (Tor), Associate Professor

Frank D.K. Roberts, MA (Cantab), MSc, PhD (Liv), Associate Professor

Margaret-Anne Storey, BSc (UVic), PhD (Simon Fraser), Associate Professor

Mantis H. M. Cheng, BMath, MMath, PhD (Wat), Assistant Professor

Yvonne Coady, BSc (Gonzaga U), MSc(Simon Fraser), PhD (Brit Col), Assistant Professor

Daniela E. Damian, BSc (Babes-Bolyai U of Cluj-Napoca), MSc (Calgary), PhD (Calgary), Assistant Professor

Sudhakar N.M. Ganti, BTech (JNTU), MTech (IIT), PhD (U of Ottawa), Assistant Professor

Daniel M. Germán, BS (UPIICSA/IPN), MS (Coll of William and Mary), PhD (Wat), Assistant Professor

Jianping Pan, BE (Southeast Univ., Nanjing), PhD (Southeast Univ., Nanjing), Assistant Professor

Venkatesh Srinivasan, BE (Birla Inst. of Technology), MSc (Birla Inst. of Technology), PhD (Tata Inst. of Fundamental Research), Assistant Professor

Ulrike Stege, Dipl Math (Albert-Ludwigs-Universität Freiburg), PhD (ETH Zürich), Assistant Professor

Alex Thomo, BSc (U of Piraeus), MSc (Concordia), PhD (Concordia) Assistant Professor

Melanie Tory, BSc (UBC), PhD (SFU), Assistant Professor

George Tzanetakis, BSE (U of Crete), MA (Princeton), PhD (Princeton), Assistant Professor

Kui Wu, BSc (Wuhan), MEng (Wuhan), PhD (Alberta), Assistant Professor

Jason Corless, BSc, MSc (UVic), Senior Instructor

LillAnne Jackson, BSc (Alberta), MSc (Lethbridge)

Mary Sanseverino, BSc, MSc (UVic), Senior Instructor

Michael Zastre, BSc (SFU), MSc, PhD (UVic), Senior Instructor and Director of Undergraduate Studies

Jillian Aschenbrenner, Programmer Analyst

Marguerite E. Casey, BSc, MEd (UVic), Co-operative Education Coordinator (Engineering & Computer Science/Math Co-op)

Bette Bultena, BSc, MSc (UVic), Senior Laboratory Instructor

Susan Fiddler, BMus (UVic), Placement Coordinator, Co-operative Education Co-ordinator (Engineering & Computer Science/Math Co-op)

Marilee V. Garrett, BA (Brown), MSc (UVic), Co-operative Education Co-ordinator (Engineering & Computer Science/Math Co-op)

William F. Gorman, BA (Queen's), Laboratory Instructor

Jane Guy, BSc (U of London), MSc (U of Zimbabwe), Academic Advisor

Victoria Li, BSc (Wuhan), MSc (Simon Fraser), Senior Laboratory Instructor

Susanne Reiser, Administrative Officer

Anissa St. Pierre, BSc (UVic), Executive Assistant, Enrolment Program

Paul Stead, Senior Programmer Analyst

Robert Taylor, Senior Systems Administrator

Allan Trumpour, BSc (UVic), Senior Programmer Analyst

Kathryn Wilson, BA (Emily Carr Inst. of Art and Design), Programmer Analyst

Visiting, Limited Term, Adjunct and Cross-Listed Appointments

Ian Barrodale, BSc (Wales), MA (Brit Col), PhD (Liv), Adjunct Professor (2005-08)

Ernest J.H. Chang, BSc (Man), MD (UBC), MMath (Wat), PhD (Tor), Adjunct Associate Professor (2004-07)

Maurice Danard, BA (Brit Col), MA (Tor), PhD (Chic), Adjunct Professor (2005-08)

Brian Gaines, BA (Trinity), MA, PhD (Cambridge), Adjunct Professor (2004-07)

David G. Goodenough, BSc (Brit Col), MSc, PhD (Tor), Adjunct Professor (2005-08)

Benjamin Jung, PhD (Trinity College, Dublin), Cross-listed Assistant Professor (2005-07)

Jacqueline Rice, BSc, MSc, PhD (UVic), Adjunct Assistant Professor (2004-07)

Dominique Roelants van Baronaigien, BSc, MSc, PhD (UVic), Adjunct Associate Professor (2003-06)

W. Andrew Schloss, BA (Bennington Coll), PhD (Stan), Cross-listed Associate Professor (2004-06)

Janice Singer, BA (Calif, San Diego), PhD (Pittsburgh), Adjunct Professor (2004-07)

Pauline van den Driessche, PhD (Wales), Cross-listed Professor (2004-06)

Maarten van Emden, MSc (T. H. Delft), PhD (Amsterdam), Adjunct Professor (2003-06)

Peter Walsh, BSc, MSc (Univ Coll, Cork), PhD (UVic), Adjunct Assistant Professor (2004-07)

Computer Science Programs

Undergraduate Programs

The Department of Computer Science offers the following programs leading to the degree of Bachelor of Science:

- Major and Honours in Computer Science
- Major in Computer Science (Software Engineering Option)
- Major in Computer Science (Business Option)
- Major in Computer Science (Mechatronics and Embedded Systems Option)
- Major in Computer Science (Bioinformatics Option)

Students who plan to pursue one of these programs and who meet the qualifications set out below should apply to the Undergraduate Admissions Office and should indicate that they wish to register in the Faculty of Engineering for their first year of study.

Students may complete a combined degree program in the following fields:

- Computer Science and Mathematics
- Computer Science and Statistics
- Physics and Computer Science
- Health Information Science and Computer Science
- Visual Arts and Computer Science
- Psychology and Computer Science
- Geography and Computer Science (Geomatics)
- Music and Computer Science

Students wishing to complete one of the combined degree programs in Computer Science and Mathematics, Statistics or Physics will normally register in the Faculty of Science for their first year.

Students wishing to complete a combined degree program in Health Information Science and Computer Science will normally register in the Faculty of Human and Social Development.

Students wishing to complete a combined degree program in Visual Arts and Computer Science will normally register in the Faculty of Fine Arts.

Since students wishing to complete a combined degree program in Music and Computer Science are admitted into this program in the Faculty of Fine Arts at the end of first year, applicants will normally register in first year in one of the following faculties/programs for which they meet the admission requirements: Humanities, Science, Social Sciences, Engineering or Fine Arts.

Students wishing to complete a combined degree program in Psychology and Computer Science, or Geography and Computer Science will normally register in the Faculty of Social Sciences.

The Department also offers the following General degree programs:

- BSc General in Computer Science
- BA General in Computer Science

Students wishing to complete a General degree in Computer Science will normally register in the faculty offering the second specialization area of their General degree in their first year.

Students may also complete a Minor in Computer Science.

Graduate Programs

The Department of Computer Science offers the following graduate degrees: MA, MSc, PhD. For information, please see the UVic Graduate Calendar.

Academic Advice

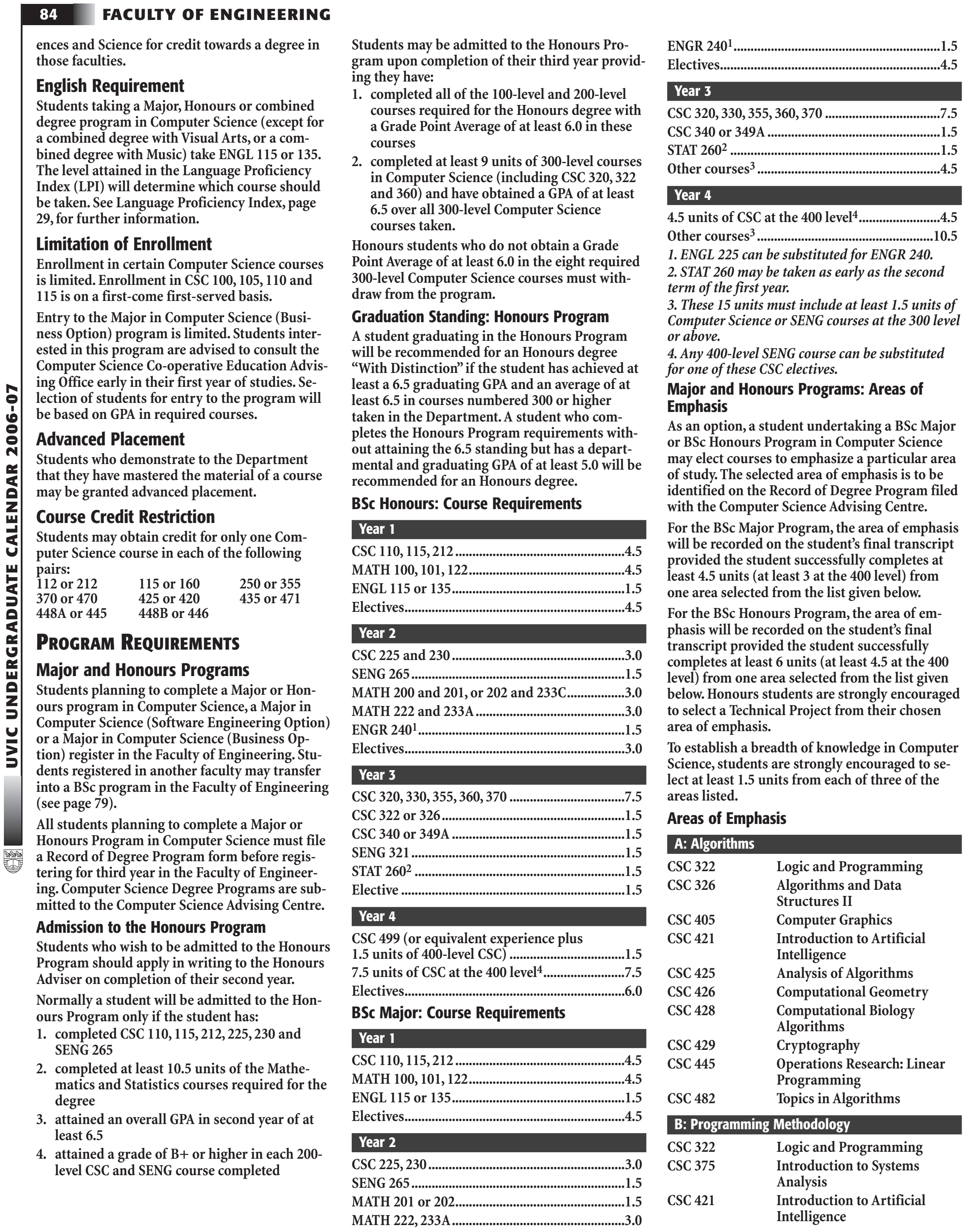
Students considering enrollment in a combined BSc in Computer Science should seek academic advice from the Advising Centre for the Faculties of Humanities, Science and Social Sciences, or the Department of Computer Science. Students considering or enrolled in a Major or Honours Program in Computer Science should seek academic advice through the Computer Science Advising Centre. Students planning to complete a Major Program in Computer Science (Business Option) should consult the Computer Science Co-operative Education Advising Office before completion of their first term of studies.

Students from outside British Columbia and students transferring from other post-secondary institutions must consult the Department before enrolling in any Computer Science course.

Availability of Courses to Students in Other Faculties

All undergraduate courses offered by the Department of Computer Science may be taken by students in the Faculties of Humanities, Social Sci-

Computer Science Undergraduate Programs, continued



ences and Science for credit towards a degree in those faculties.

English Requirement

Students taking a Major, Honours or combined degree program in Computer Science (except for a combined degree with Visual Arts, or a combined degree with Music) take ENGL 115 or 135. The level attained in the Language Proficiency Index (LPI) will determine which course should be taken. See Language Proficiency Index, page 29, for further information.

Limitation of Enrollment

Enrollment in certain Computer Science courses is limited. Enrollment in CSC 100, 105, 110 and 115 is on a first-come first-served basis.

Entry to the Major in Computer Science (Business Option) program is limited. Students interested in this program are advised to consult the Computer Science Co-operative Education Advising Office early in their first year of studies. Selection of students for entry to the program will be based on GPA in required courses.

Advanced Placement

Students who demonstrate to the Department that they have mastered the material of a course may be granted advanced placement.

Course Credit Restriction

Students may obtain credit for only one Computer Science course in each of the following pairs:

112 or 212	115 or 160	250 or 355
370 or 470	425 or 420	435 or 471
448A or 445	448B or 446	

Program Requirements

Major and Honours Programs

Students planning to complete a Major or Honours program in Computer Science, a Major in Computer Science (Software Engineering Option) or a Major in Computer Science (Business Option) register in the Faculty of Engineering. Students registered in another faculty may transfer into a BSc program in the Faculty of Engineering (see page 79).

All students planning to complete a Major or Honours Program in Computer Science must file a Record of Degree Program form before registering for third year in the Faculty of Engineering. Computer Science Degree Programs are submitted to the Computer Science Advising Centre.

Admission to the Honours Program

Students who wish to be admitted to the Honours Program should apply in writing to the Honours Adviser on completion of their second year.

Normally a student will be admitted to the Honours Program only if the student has:

1. completed CSC 110, 115, 212, 225, 230 and SENG 265
2. completed at least 10.5 units of the Mathematics and Statistics courses required for the degree
3. attained an overall GPA in second year of at least 6.5
4. attained a grade of B+ or higher in each 200-level CSC and SENG course completed

Students may be admitted to the Honours Program upon completion of their third year providing they have:

1. completed all of the 100-level and 200-level courses required for the Honours degree with a Grade Point Average of at least 6.0 in these courses
2. completed at least 9 units of 300-level courses in Computer Science (including CSC 320, 322 and 360) and have obtained a GPA of at least 6.5 over all 300-level Computer Science courses taken.

Honours students who do not obtain a Grade Point Average of at least 6.0 in the eight required 300-level Computer Science courses must withdraw from the program.

Graduation Standing: Honours Program

A student graduating in the Honours Program will be recommended for an Honours degree "With Distinction" if the student has achieved at least a 6.5 graduating GPA and an average of at least 6.5 in courses numbered 300 or higher taken in the Department. A student who completes the Honours Program requirements without attaining the 6.5 standing but has a departmental and graduating GPA of at least 5.0 will be recommended for an Honours degree.

BSc Honours: Course Requirements

Year 1	
CSC 110, 115, 212	4.5
MATH 100, 101, 122	4.5
ENGL 115 or 135	1.5
Electives	4.5
Year 2	
CSC 225 and 230	3.0
SENG 265	1.5
MATH 200 and 201, or 202 and 233C	3.0
MATH 222 and 233A	3.0
ENGR 240[1]	1.5
Electives	3.0
Year 3	
CSC 320, 330, 355, 360, 370	7.5
CSC 322 or 326	1.5
CSC 340 or 349A	1.5
SENG 321	1.5
STAT 260[2]	1.5
Elective	1.5
Year 4	
CSC 499 (or equivalent experience plus 1.5 units of 400-level CSC)	1.5
7.5 units of CSC at the 400 level[4]	7.5
Electives	6.0

BSc Major: Course Requirements

Year 1	
CSC 110, 115, 212	4.5
MATH 100, 101, 122	4.5
ENGL 115 or 135	1.5
Electives	4.5
Year 2	
CSC 225, 230	3.0
SENG 265	1.5
MATH 201 or 202	1.5
MATH 222, 233A	3.0
ENGR 240[1]	1.5
Electives	4.5
Year 3	
CSC 320, 330, 355, 360, 370	7.5
CSC 340 or 349A	1.5
STAT 260[2]	1.5
Other courses[3]	4.5
Year 4	
4.5 units of CSC at the 400 level[4]	4.5
Other courses[3]	10.5

[1] ENGL 225 can be substituted for ENGR 240.

[2] STAT 260 may be taken as early as the second term of the first year.

[3] These 15 units must include at least 1.5 units of Computer Science or SENG courses at the 300 level or above.

[4] Any 400-level SENG course can be substituted for one of these CSC electives.

Major and Honours Programs: Areas of Emphasis

As an option, a student undertaking a BSc Major or BSc Honours Program in Computer Science may elect courses to emphasize a particular area of study. The selected area of emphasis is to be identified on the Record of Degree Program filed with the Computer Science Advising Centre.

For the BSc Major Program, the area of emphasis will be recorded on the student's final transcript provided the student successfully completes at least 4.5 units (at least 3 at the 400 level) from one area selected from the list given below.

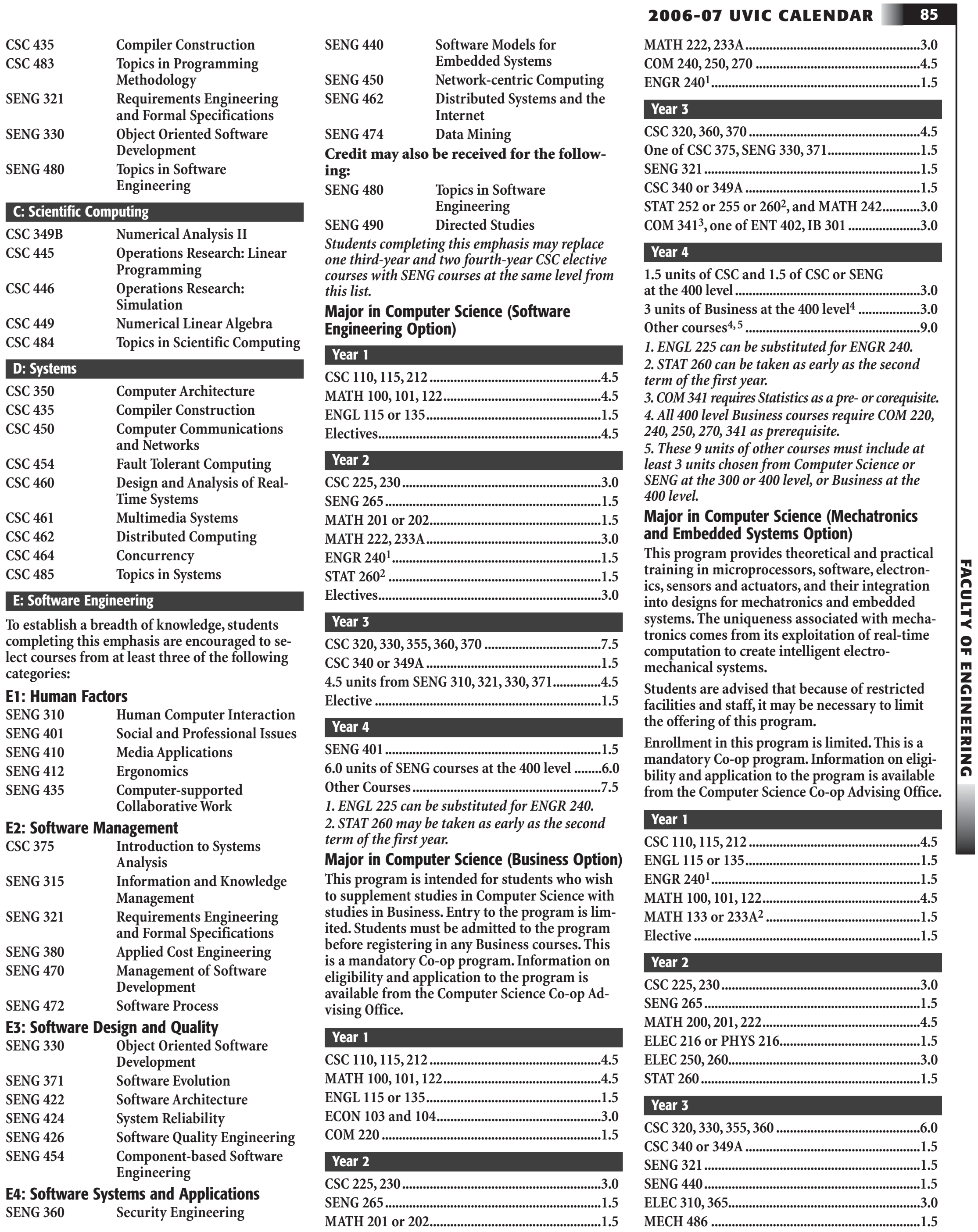
For the BSc Honours Program, the area of emphasis will be recorded on the student's final transcript provided the student successfully completes at least 6 units (at least 4.5 at the 400 level) from one area selected from the list given below. Honours students are strongly encouraged to select a Technical Project from their chosen area of emphasis.

To establish a breadth of knowledge in Computer Science, students are strongly encouraged to select at least 1.5 units from each of three of the areas listed.

Areas of Emphasis

A: Algorithms	
CSC 322	Logic and Programming
CSC 326	Algorithms and Data Structures II
CSC 405	Computer Graphics
CSC 421	Introduction to Artificial Intelligence
CSC 425	Analysis of Algorithms
CSC 426	Computational Geometry
CSC 428	Computational Biology Algorithms
CSC 429	Cryptography
CSC 445	Operations Research: Linear Programming
CSC 482	Topics in Algorithms
B: Programming Methodology	
CSC 322	Logic and Programming
CSC 375	Introduction to Systems Analysis
CSC 421	Introduction to Artificial Intelligence

Computer Science Undergraduate Programs, continued



CSC 435	Compiler Construction
CSC 483	Topics in Programming Methodology
SENG 321	Requirements Engineering and Formal Specifications
SENG 330	Object Oriented Software Development
SENG 480	Topics in Software Engineering

C: Scientific Computing

CSC 349B	Numerical Analysis II
CSC 445	Operations Research: Linear Programming
CSC 446	Operations Research: Simulation
CSC 449	Numerical Linear Algebra
CSC 484	Topics in Scientific Computing

D: Systems

CSC 350	Computer Architecture
CSC 435	Compiler Construction
CSC 450	Computer Communications and Networks
CSC 454	Fault Tolerant Computing
CSC 460	Design and Analysis of Real-Time Systems
CSC 461	Multimedia Systems
CSC 462	Distributed Computing
CSC 464	Concurrency
CSC 485	Topics in Systems

E: Software Engineering

To establish a breadth of knowledge, students completing this emphasis are encouraged to select courses from at least three of the following categories:

E1: Human Factors

SENG 310	Human Computer Interaction
SENG 401	Social and Professional Issues
SENG 410	Media Applications
SENG 412	Ergonomics
SENG 435	Computer-supported Collaborative Work

E2: Software Management

CSC 375	Introduction to Systems Analysis
SENG 315	Information and Knowledge Management
SENG 321	Requirements Engineering and Formal Specifications
SENG 380	Applied Cost Engineering
SENG 470	Management of Software Development
SENG 472	Software Process

E3: Software Design and Quality

SENG 330	Object Oriented Software Development
SENG 371	Software Evolution
SENG 422	Software Architecture
SENG 424	System Reliability
SENG 426	Software Quality Engineering
SENG 454	Component-based Software Engineering

E4: Software Systems and Applications

SENG 360	Security Engineering
SENG 440	Software Models for Embedded Systems
SENG 450	Network-centric Computing
SENG 462	Distributed Systems and the Internet
SENG 474	Data Mining

Credit may also be received for the following:

SENG 480	Topics in Software Engineering
SENG 490	Directed Studies

Students completing this emphasis may replace one third-year and two fourth-year CSC elective courses with SENG courses at the same level from this list.

Major in Computer Science (Software Engineering Option)

Year 1	
CSC 110, 115, 212	4.5
MATH 100, 101, 122	4.5
ENGL 115 or 135	1.5
Electives	4.5

Year 2	
CSC 225, 230	3.0
SENG 265	1.5
MATH 201 or 202	1.5
MATH 222, 233A	3.0
ENGR 240[1]	1.5
STAT 260[2]	1.5
Electives	3.0

Year 3	
CSC 320, 330, 355, 360, 370	7.5
CSC 340 or 349A	1.5
4.5 units from SENG 310, 321, 330, 371	4.5
Elective	1.5

Year 4	
SENG 401	1.5
6.0 units of SENG courses at the 400 level	6.0
Other Courses	7.5

1. ENGL 225 can be substituted for ENGR 240.
2. STAT 260 may be taken as early as the second term of the first year.

Major in Computer Science (Business Option)

This program is intended for students who wish to supplement studies in Computer Science with studies in Business. Entry to the program is limited. Students must be admitted to the program before registering in any Business courses. This is a mandatory Co-op program. Information on eligibility and application to the program is available from the Computer Science Co-op Advising Office.

Year 1	
CSC 110, 115, 212	4.5
MATH 100, 101, 122	4.5
ENGL 115 or 135	1.5
ECON 103 and 104	3.0
COM 220	1.5

Year 2	
CSC 225, 230	3.0
SENG 265	1.5
MATH 201 or 202	1.5
MATH 222, 233A	3.0
COM 240, 250, 270	4.5
ENGR 240[1]	1.5

Year 3	
CSC 320, 360, 370	4.5
One of CSC 375, SENG 330, 371	1.5
SENG 321	1.5
CSC 340 or 349A	1.5
STAT 252 or 255 or 260[2], and MATH 242	3.0
COM 341[3], one of ENT 402, IB 301	3.0

Year 4	
1.5 units of CSC and 1.5 of CSC or SENG at the 400 level	3.0
3 units of Business at the 400 level[4]	3.0
Other courses[4, 5]	9.0

1. ENGL 225 can be substituted for ENGR 240.
2. STAT 260 can be taken as early as the second term of the first year.
3. COM 341 requires Statistics as a pre- or corequisite.
4. All 400 level Business courses require COM 220, 240, 250, 270, 341 as prerequisite.
5. These 9 units of other courses must include at least 3 units chosen from Computer Science or SENG at the 300 or 400 level, or Business at the 400 level.

Major in Computer Science (Mechatronics and Embedded Systems Option)

This program provides theoretical and practical training in microprocessors, software, electronics, sensors and actuators, and their integration into designs for mechatronics and embedded systems. The uniqueness associated with mechatronics comes from its exploitation of real-time computation to create intelligent electro-mechanical systems.

Students are advised that because of restricted facilities and staff, it may be necessary to limit the offering of this program.

Enrollment in this program is limited. This is a mandatory Co-op program. Information on eligibility and application to the program is available from the Computer Science Co-op Advising Office.

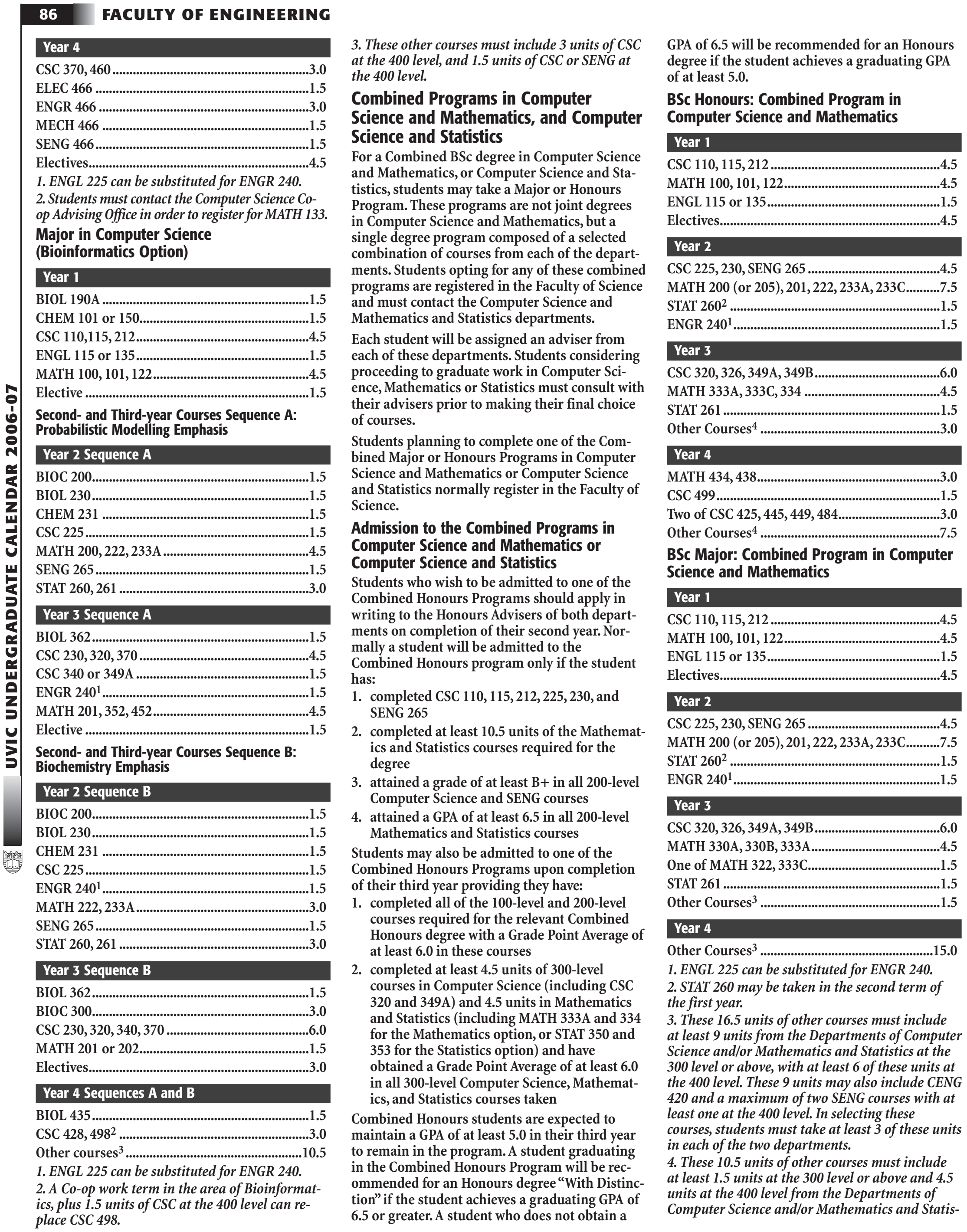
Year 1	
CSC 110, 115, 212	4.5
ENGL 115 or 135	1.5
ENGR 240[1]	1.5
MATH 100, 101, 122	4.5
MATH 133 or 233A[2]	1.5
Elective	1.5

Year 2	
CSC 225, 230	3.0
SENG 265	1.5
MATH 200, 201, 222	4.5
ELEC 216 or PHYS 216	1.5
ELEC 250, 260	3.0
STAT 260	1.5

Year 3	
CSC 320, 330, 355, 360	6.0
CSC 340 or 349A	1.5
SENG 321	1.5
SENG 440	1.5
ELEC 310, 365	3.0
MECH 486	1.5

FACULTY OF ENGINEERING

Computer Science Undergraduate Programs, continued



Year 4	
CSC 370, 460	3.0
ELEC 466	1.5
ENGR 466	3.0
MECH 466	1.5
SENG 466	1.5
Electives	4.5

1. ENGL 225 can be substituted for ENGR 240.

2. Students must contact the Computer Science Co-op Advising Office in order to register for MATH 133.

Major in Computer Science (Bioinformatics Option)

Year 1	
BIOL 190A	1.5
CHEM 101 or 150	1.5
CSC 110,115, 212	4.5
ENGL 115 or 135	1.5
MATH 100, 101, 122	4.5
Elective	1.5

Second- and Third-year Courses Sequence A: Probabilistic Modelling Emphasis

Year 2 Sequence A	
BIOC 200	1.5
BIOL 230	1.5
CHEM 231	1.5
CSC 225	1.5
MATH 200, 222, 233A	4.5
SENG 265	1.5
STAT 260, 261	3.0

Year 3 Sequence A	
BIOL 362	1.5
CSC 230, 320, 370	4.5
CSC 340 or 349A	1.5
ENGR 240[1]	1.5
MATH 201, 352, 452	4.5
Elective	1.5

Second- and Third-year Courses Sequence B: Biochemistry Emphasis

Year 2 Sequence B	
BIOC 200	1.5
BIOL 230	1.5
CHEM 231	1.5
CSC 225	1.5
ENGR 240[1]	1.5
MATH 222, 233A	3.0
SENG 265	1.5
STAT 260, 261	3.0

Year 3 Sequence B	
BIOL 362	1.5
BIOC 300	3.0
CSC 230, 320, 340, 370	6.0
MATH 201 or 202	1.5
Electives	3.0

Year 4 Sequences A and B	
BIOL 435	1.5
CSC 428, 498[2]	3.0
Other courses[3]	10.5

1. ENGL 225 can be substituted for ENGR 240.

2. A Co-op work term in the area of Bioinformatics, plus 1.5 units of CSC at the 400 level can replace CSC 498.

3. These other courses must include 3 units of CSC at the 400 level, and 1.5 units of CSC or SENG at the 400 level.

Combined Programs in Computer Science and Mathematics, and Computer Science and Statistics

For a Combined BSc degree in Computer Science and Mathematics, or Computer Science and Statistics, students may take a Major or Honours Program. These programs are not joint degrees in Computer Science and Mathematics, but a single degree program composed of a selected combination of courses from each of the departments. Students opting for any of these combined programs are registered in the Faculty of Science and must contact the Computer Science and Mathematics and Statistics departments.

Each student will be assigned an adviser from each of these departments. Students considering proceeding to graduate work in Computer Science, Mathematics or Statistics must consult with their advisers prior to making their final choice of courses.

Students planning to complete one of the Combined Major or Honours Programs in Computer Science and Mathematics or Computer Science and Statistics normally register in the Faculty of Science.

Admission to the Combined Programs in Computer Science and Mathematics or Computer Science and Statistics

Students who wish to be admitted to one of the Combined Honours Programs should apply in writing to the Honours Advisers of both departments on completion of their second year. Normally a student will be admitted to the Combined Honours program only if the student has:

1. completed CSC 110, 115, 212, 225, 230, and SENG 265
2. completed at least 10.5 units of the Mathematics and Statistics courses required for the degree
3. attained a grade of at least B+ in all 200-level Computer Science and SENG courses
4. attained a GPA of at least 6.5 in all 200-level Mathematics and Statistics courses

Students may also be admitted to one of the Combined Honours Programs upon completion of their third year providing they have:

1. completed all of the 100-level and 200-level courses required for the relevant Combined Honours degree with a Grade Point Average of at least 6.0 in these courses
2. completed at least 4.5 units of 300-level courses in Computer Science (including CSC 320 and 349A) and 4.5 units in Mathematics and Statistics (including MATH 333A and 334 for the Mathematics option, or STAT 350 and 353 for the Statistics option) and have obtained a Grade Point Average of at least 6.0 in all 300-level Computer Science, Mathematics, and Statistics courses taken

Combined Honours students are expected to maintain a GPA of at least 5.0 in their third year to remain in the program. A student graduating in the Combined Honours Program will be recommended for an Honours degree "With Distinction" if the student achieves a graduating GPA of 6.5 or greater. A student who does not obtain a GPA of 6.5 will be recommended for an Honours degree if the student achieves a graduating GPA of at least 5.0.

BSc Honours: Combined Program in Computer Science and Mathematics

Year 1	
CSC 110, 115, 212	4.5
MATH 100, 101, 122	4.5
ENGL 115 or 135	1.5
Electives	4.5

Year 2	
CSC 225, 230, SENG 265	4.5
MATH 200 (or 205), 201, 222, 233A, 233C	7.5
STAT 260[2]	1.5
ENGR 240[1]	1.5

Year 3	
CSC 320, 326, 349A, 349B	6.0
MATH 333A, 333C, 334	4.5
STAT 261	1.5
Other Courses[4]	3.0

Year 4	
MATH 434, 438	3.0
CSC 499	1.5
Two of CSC 425, 445, 449, 484	3.0
Other Courses[4]	7.5

BSc Major: Combined Program in Computer Science and Mathematics

Year 1	
CSC 110, 115, 212	4.5
MATH 100, 101, 122	4.5
ENGL 115 or 135	1.5
Electives	4.5

Year 2	
CSC 225, 230, SENG 265	4.5
MATH 200 (or 205), 201, 222, 233A, 233C	7.5
STAT 260[2]	1.5
ENGR 240[1]	1.5

Year 3	
CSC 320, 326, 349A, 349B	6.0
MATH 330A, 330B, 333A	4.5
One of MATH 322, 333C	1.5
STAT 261	1.5
Other Courses[3]	1.5

Year 4	
Other Courses[3]	15.0

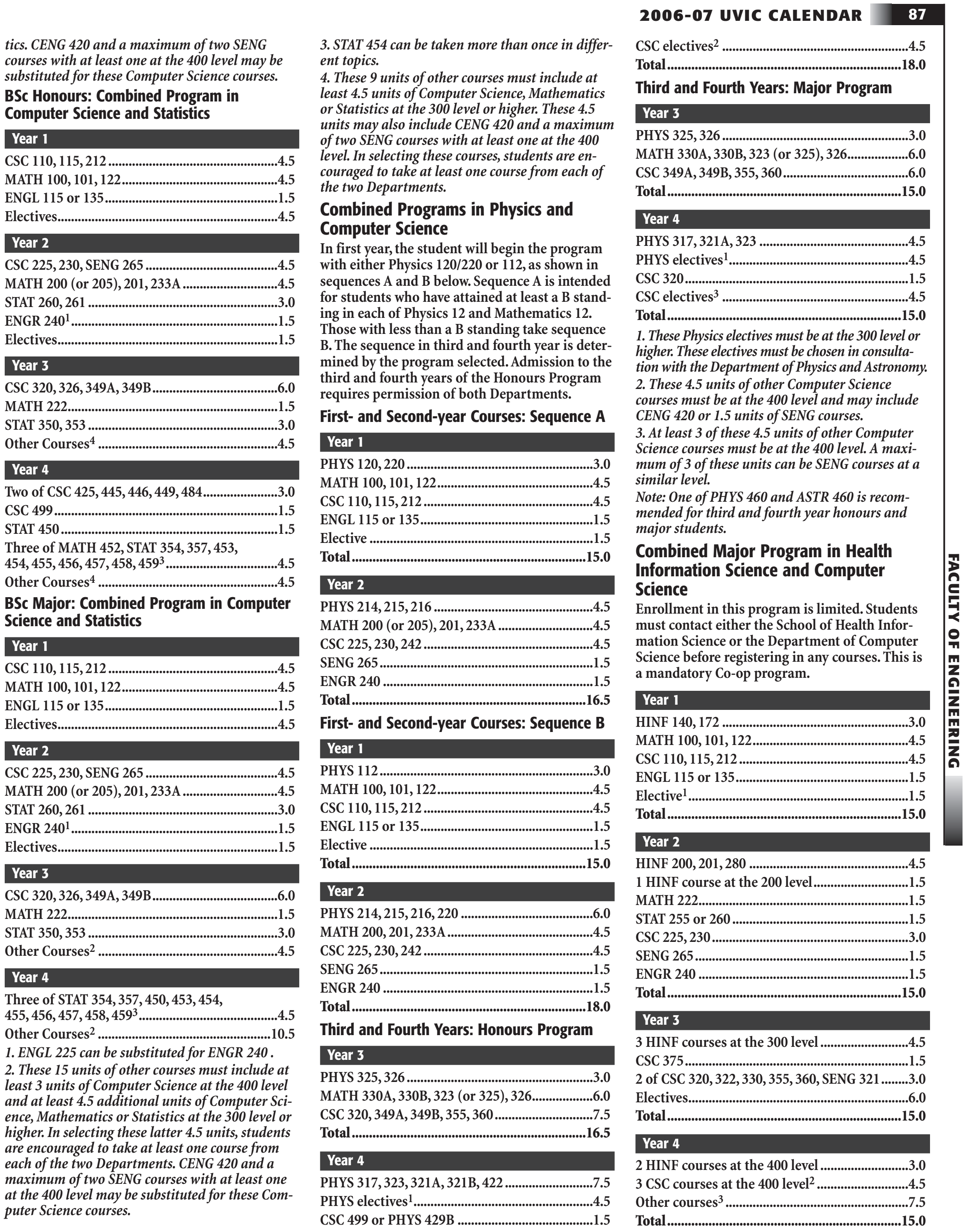
1. ENGL 225 can be substituted for ENGR 240.

2. STAT 260 may be taken in the second term of the first year.

3. These 16.5 units of other courses must include at least 9 units from the Departments of Computer Science and/or Mathematics and Statistics at the 300 level or above, with at least 6 of these units at the 400 level. These 9 units may also include CENG 420 and a maximum of two SENG courses with at least one at the 400 level. In selecting these courses, students must take at least 3 of these units in each of the two departments.

4. These 10.5 units of other courses must include at least 1.5 units at the 300 level or above and 4.5 units at the 400 level from the Departments of Computer Science and/or Mathematics and Statis-

Computer Science Undergraduate Programs, continued



tics. CENG 420 and a maximum of two SENG courses with at least one at the 400 level may be substituted for these Computer Science courses.

BSc Honours: Combined Program in Computer Science and Statistics

Year 1	
CSC 110, 115, 212	4.5
MATH 100, 101, 122	4.5
ENGL 115 or 135	1.5
Electives	4.5

Year 2	
CSC 225, 230, SENG 265	4.5
MATH 200 (or 205), 201, 233A	4.5
STAT 260, 261	3.0
ENGR 240[1]	1.5
Electives	1.5

Year 3	
CSC 320, 326, 349A, 349B	6.0
MATH 222	1.5
STAT 350, 353	3.0
Other Courses[4]	4.5

Year 4	
Two of CSC 425, 445, 446, 449, 484	3.0
CSC 499	1.5
STAT 450	1.5
Three of MATH 452, STAT 354, 357, 453, 454, 455, 456, 457, 458, 459[3]	4.5
Other Courses[4]	4.5

BSc Major: Combined Program in Computer Science and Statistics

Year 1	
CSC 110, 115, 212	4.5
MATH 100, 101, 122	4.5
ENGL 115 or 135	1.5
Electives	4.5

Year 2	
CSC 225, 230, SENG 265	4.5
MATH 200 (or 205), 201, 233A	4.5
STAT 260, 261	3.0
ENGR 240[1]	1.5
Electives	1.5

Year 3	
CSC 320, 326, 349A, 349B	6.0
MATH 222	1.5
STAT 350, 353	3.0
Other Courses[2]	4.5

Year 4	
Three of STAT 354, 357, 450, 453, 454, 455, 456, 457, 458, 459[3]	4.5
Other Courses[2]	10.5

[1] ENGL 225 can be substituted for ENGR 240 .

[2] These 15 units of other courses must include at least 3 units of Computer Science at the 400 level and at least 4.5 additional units of Computer Science, Mathematics or Statistics at the 300 level or higher. In selecting these latter 4.5 units, students are encouraged to take at least one course from each of the two Departments. CENG 420 and a maximum of two SENG courses with at least one at the 400 level may be substituted for these Computer Science courses.

[3] STAT 454 can be taken more than once in different topics.

[4] These 9 units of other courses must include at least 4.5 units of Computer Science, Mathematics or Statistics at the 300 level or higher. These 4.5 units may also include CENG 420 and a maximum of two SENG courses with at least one at the 400 level. In selecting these courses, students are encouraged to take at least one course from each of the two Departments.

Combined Programs in Physics and Computer Science

In first year, the student will begin the program with either Physics 120/220 or 112, as shown in sequences A and B below. Sequence A is intended for students who have attained at least a B standing in each of Physics 12 and Mathematics 12. Those with less than a B standing take sequence B. The sequence in third and fourth year is determined by the program selected. Admission to the third and fourth years of the Honours Program requires permission of both Departments.

First- and Second-year Courses: Sequence A

Year 1	
PHYS 120, 220	3.0
MATH 100, 101, 122	4.5
CSC 110, 115, 212	4.5
ENGL 115 or 135	1.5
Elective	1.5
Total	**15.0**

Year 2	
PHYS 214, 215, 216	4.5
MATH 200 (or 205), 201, 233A	4.5
CSC 225, 230, 242	4.5
SENG 265	1.5
ENGR 240	1.5
Total	**16.5**

First- and Second-year Courses: Sequence B

Year 1	
PHYS 112	3.0
MATH 100, 101, 122	4.5
CSC 110, 115, 212	4.5
ENGL 115 or 135	1.5
Elective	1.5
Total	**15.0**

Year 2	
PHYS 214, 215, 216, 220	6.0
MATH 200, 201, 233A	4.5
CSC 225, 230, 242	4.5
SENG 265	1.5
ENGR 240	1.5
Total	**18.0**

Third and Fourth Years: Honours Program

Year 3	
PHYS 325, 326	3.0
MATH 330A, 330B, 323 (or 325), 326	6.0
CSC 320, 349A, 349B, 355, 360	7.5
Total	**16.5**

Year 4	
PHYS 317, 323, 321A, 321B, 422	7.5
PHYS electives[1]	4.5
CSC 499 or PHYS 429B	1.5
CSC electives[2]	4.5
Total	**18.0**

Third and Fourth Years: Major Program

Year 3	
PHYS 325, 326	3.0
MATH 330A, 330B, 323 (or 325), 326	6.0
CSC 349A, 349B, 355, 360	6.0
Total	**15.0**

Year 4	
PHYS 317, 321A, 323	4.5
PHYS electives[1]	4.5
CSC 320	1.5
CSC electives[3]	4.5
Total	**15.0**

[1] These Physics electives must be at the 300 level or higher. These electives must be chosen in consultation with the Department of Physics and Astronomy.

[2] These 4.5 units of other Computer Science courses must be at the 400 level and may include CENG 420 or 1.5 units of SENG courses.

[3] At least 3 of these 4.5 units of other Computer Science courses must be at the 400 level. A maximum of 3 of these units can be SENG courses at a similar level.

Note: One of PHYS 460 and ASTR 460 is recommended for third and fourth year honours and major students.

Combined Major Program in Health Information Science and Computer Science

Enrollment in this program is limited. Students must contact either the School of Health Information Science or the Department of Computer Science before registering in any courses. This is a mandatory Co-op program.

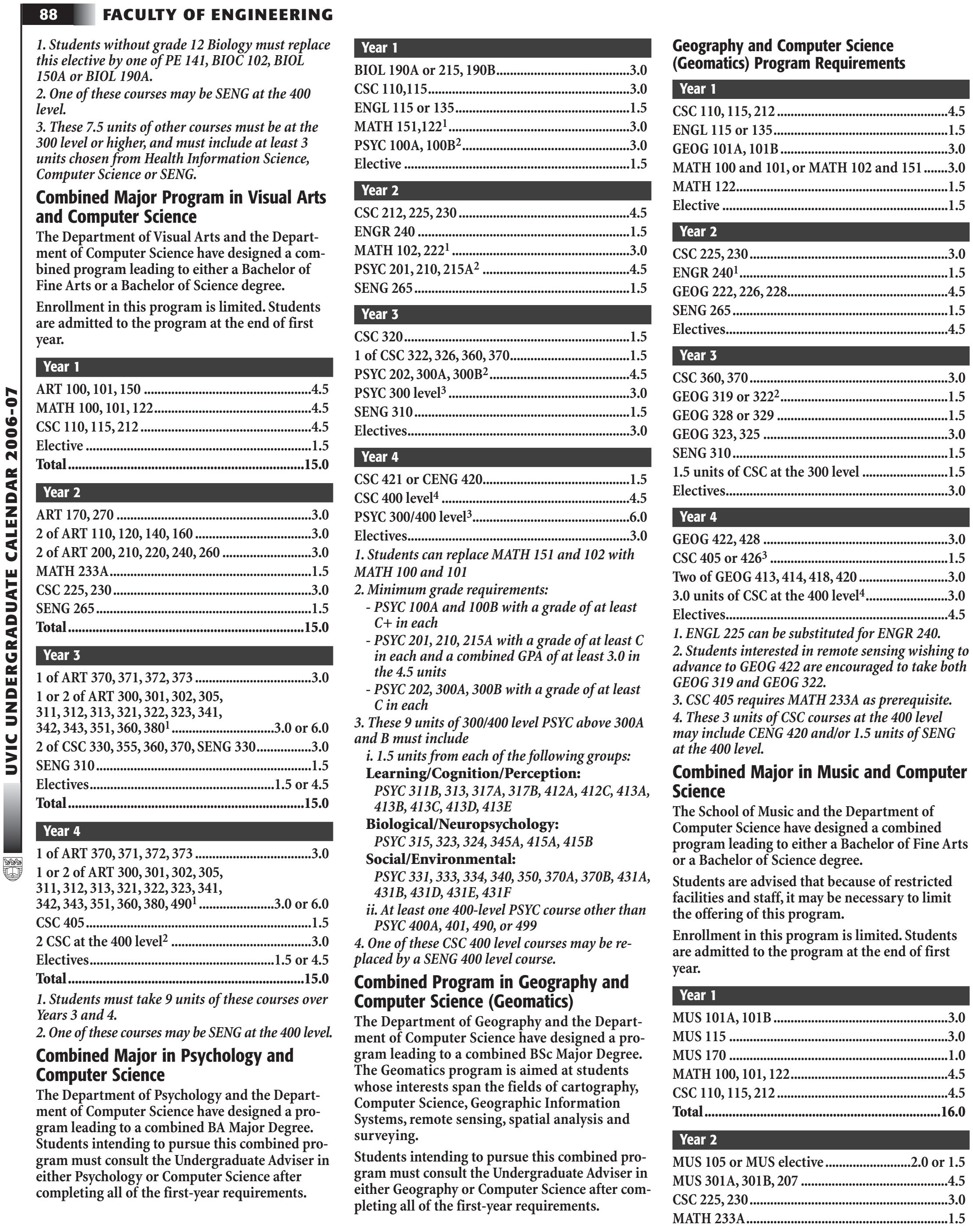
Year 1	
HINF 140, 172	3.0
MATH 100, 101, 122	4.5
CSC 110, 115, 212	4.5
ENGL 115 or 135	1.5
Elective[1]	1.5
Total	**15.0**

Year 2	
HINF 200, 201, 280	4.5
1 HINF course at the 200 level	1.5
MATH 222	1.5
STAT 255 or 260	1.5
CSC 225, 230	3.0
SENG 265	1.5
ENGR 240	1.5
Total	**15.0**

Year 3	
3 HINF courses at the 300 level	4.5
CSC 375	1.5
2 of CSC 320, 322, 330, 355, 360, SENG 321	3.0
Electives	6.0
Total	**15.0**

Year 4	
2 HINF courses at the 400 level	3.0
3 CSC courses at the 400 level[2]	4.5
Other courses[3]	7.5
Total	**15.0**

Computer Science Undergraduate Programs, continued



1. Students without grade 12 Biology must replace this elective by one of PE 141, BIOC 102, BIOL 150A or BIOL 190A.

2. One of these courses may be SENG at the 400 level.

3. These 7.5 units of other courses must be at the 300 level or higher, and must include at least 3 units chosen from Health Information Science, Computer Science or SENG.

Combined Major Program in Visual Arts and Computer Science

The Department of Visual Arts and the Department of Computer Science have designed a combined program leading to either a Bachelor of Fine Arts or a Bachelor of Science degree.

Enrollment in this program is limited. Students are admitted to the program at the end of first year.

Year 1	
ART 100, 101, 150	4.5
MATH 100, 101, 122	4.5
CSC 110, 115, 212	4.5
Elective	1.5
Total	**15.0**

Year 2	
ART 170, 270	3.0
2 of ART 110, 120, 140, 160	3.0
2 of ART 200, 210, 220, 240, 260	3.0
MATH 233A	1.5
CSC 225, 230	3.0
SENG 265	1.5
Total	**15.0**

Year 3	
1 of ART 370, 371, 372, 373	3.0
1 or 2 of ART 300, 301, 302, 305, 311, 312, 313, 321, 322, 323, 341, 342, 343, 351, 360, 380[1]	3.0 or 6.0
2 of CSC 330, 355, 360, 370, SENG 330	3.0
SENG 310	1.5
Electives	1.5 or 4.5
Total	**15.0**

Year 4	
1 of ART 370, 371, 372, 373	3.0
1 or 2 of ART 300, 301, 302, 305, 311, 312, 313, 321, 322, 323, 341, 342, 343, 351, 360, 380, 490[1]	3.0 or 6.0
CSC 405	1.5
2 CSC at the 400 level[2]	3.0
Electives	1.5 or 4.5
Total	**15.0**

1. Students must take 9 units of these courses over Years 3 and 4.

2. One of these courses may be SENG at the 400 level.

Combined Major in Psychology and Computer Science

The Department of Psychology and the Department of Computer Science have designed a program leading to a combined BA Major Degree. Students intending to pursue this combined program must consult the Undergraduate Adviser in either Psychology or Computer Science after completing all of the first-year requirements.

Year 1	
BIOL 190A or 215, 190B	3.0
CSC 110,115	3.0
ENGL 115 or 135	1.5
MATH 151,122[1]	3.0
PSYC 100A, 100B[2]	3.0
Elective	1.5

Year 2	
CSC 212, 225, 230	4.5
ENGR 240	1.5
MATH 102, 222[1]	3.0
PSYC 201, 210, 215A[2]	4.5
SENG 265	1.5

Year 3	
CSC 320	1.5
1 of CSC 322, 326, 360, 370	1.5
PSYC 202, 300A, 300B[2]	4.5
PSYC 300 level[3]	3.0
SENG 310	1.5
Electives	3.0

Year 4	
CSC 421 or CENG 420	1.5
CSC 400 level[4]	4.5
PSYC 300/400 level[3]	6.0
Electives	3.0

1. Students can replace MATH 151 and 102 with MATH 100 and 101

2. Minimum grade requirements:

- *PSYC 100A and 100B with a grade of at least C+ in each*
- *PSYC 201, 210, 215A with a grade of at least C in each and a combined GPA of at least 3.0 in the 4.5 units*
- *PSYC 202, 300A, 300B with a grade of at least C in each*

3. These 9 units of 300/400 level PSYC above 300A and B must include

i. 1.5 units from each of the following groups:

Learning/Cognition/Perception:
PSYC 311B, 313, 317A, 317B, 412A, 412C, 413A, 413B, 413C, 413D, 413E

Biological/Neuropsychology:
PSYC 315, 323, 324, 345A, 415A, 415B

Social/Environmental:
PSYC 331, 333, 334, 340, 350, 370A, 370B, 431A, 431B, 431D, 431E, 431F

ii. At least one 400-level PSYC course other than PSYC 400A, 401, 490, or 499

4. One of these CSC 400 level courses may be replaced by a SENG 400 level course.

Combined Program in Geography and Computer Science (Geomatics)

The Department of Geography and the Department of Computer Science have designed a program leading to a combined BSc Major Degree. The Geomatics program is aimed at students whose interests span the fields of cartography, Computer Science, Geographic Information Systems, remote sensing, spatial analysis and surveying.

Students intending to pursue this combined program must consult the Undergraduate Adviser in either Geography or Computer Science after completing all of the first-year requirements.

Geography and Computer Science (Geomatics) Program Requirements

Year 1	
CSC 110, 115, 212	4.5
ENGL 115 or 135	1.5
GEOG 101A, 101B	3.0
MATH 100 and 101, or MATH 102 and 151	3.0
MATH 122	1.5
Elective	1.5

Year 2	
CSC 225, 230	3.0
ENGR 240[1]	1.5
GEOG 222, 226, 228	4.5
SENG 265	1.5
Electives	4.5

Year 3	
CSC 360, 370	3.0
GEOG 319 or 322[2]	1.5
GEOG 328 or 329	1.5
GEOG 323, 325	3.0
SENG 310	1.5
1.5 units of CSC at the 300 level	1.5
Electives	3.0

Year 4	
GEOG 422, 428	3.0
CSC 405 or 426[3]	1.5
Two of GEOG 413, 414, 418, 420	3.0
3.0 units of CSC at the 400 level[4]	3.0
Electives	4.5

1. ENGL 225 can be substituted for ENGR 240.

2. Students interested in remote sensing wishing to advance to GEOG 422 are encouraged to take both GEOG 319 and GEOG 322.

3. CSC 405 requires MATH 233A as prerequisite.

4. These 3 units of CSC courses at the 400 level may include CENG 420 and/or 1.5 units of SENG at the 400 level.

Combined Major in Music and Computer Science

The School of Music and the Department of Computer Science have designed a combined program leading to either a Bachelor of Fine Arts or a Bachelor of Science degree.

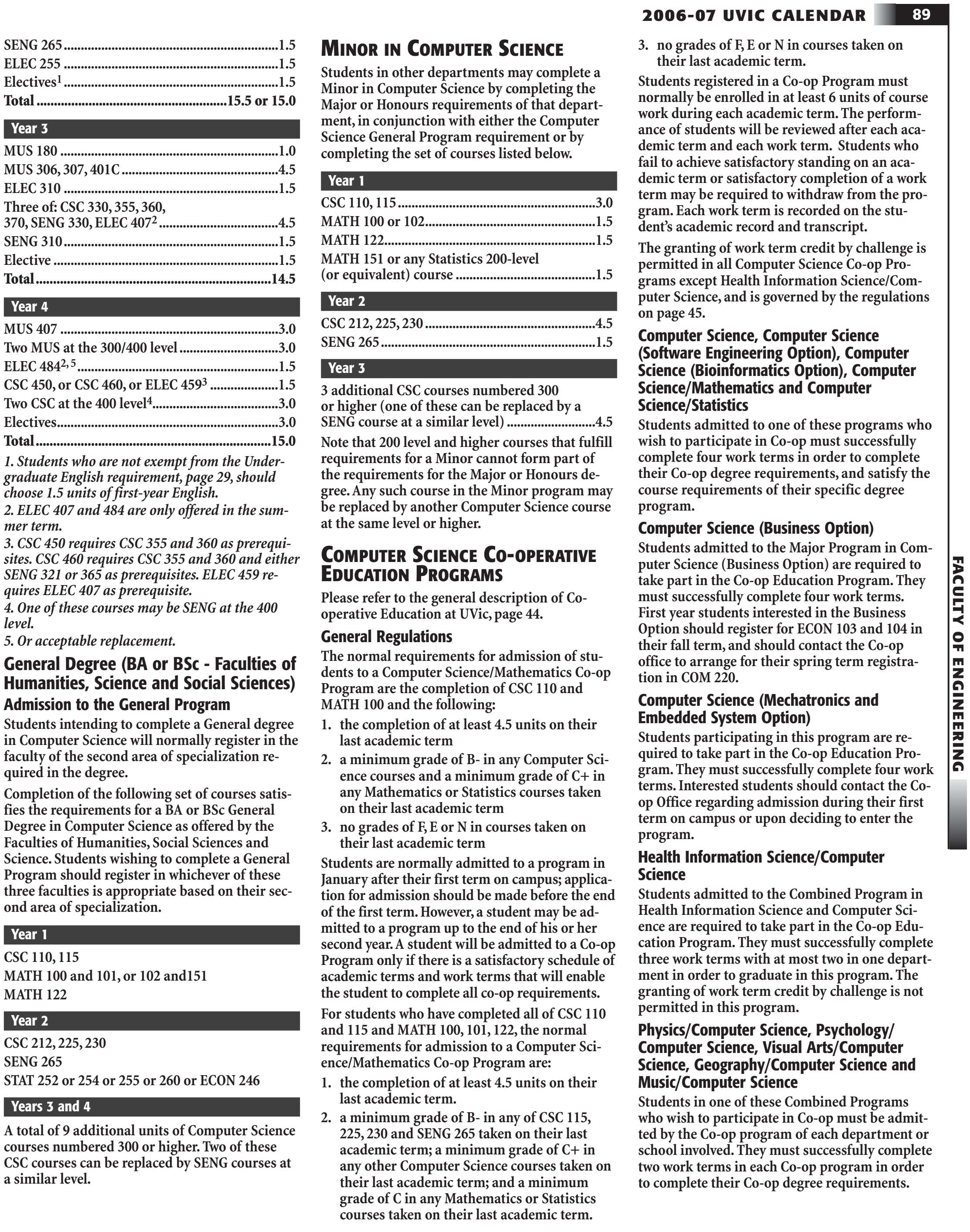
Students are advised that because of restricted facilities and staff, it may be necessary to limit the offering of this program.

Enrollment in this program is limited. Students are admitted to the program at the end of first year.

Year 1	
MUS 101A, 101B	3.0
MUS 115	3.0
MUS 170	1.0
MATH 100, 101, 122	4.5
CSC 110, 115, 212	4.5
Total	**16.0**

Year 2	
MUS 105 or MUS elective	2.0 or 1.5
MUS 301A, 301B, 207	4.5
CSC 225, 230	3.0
MATH 233A	1.5

Computer Science Undergraduate Programs, continued



SENG 265	1.5
ELEC 255	1.5
Electives[1]	1.5
Total	**15.5 or 15.0**

Year 3	
MUS 180	1.0
MUS 306, 307, 401C	4.5
ELEC 310	1.5
Three of: CSC 330, 355, 360, 370, SENG 330, ELEC 407[2]	4.5
SENG 310	1.5
Elective	1.5
Total	**14.5**

Year 4	
MUS 407	3.0
Two MUS at the 300/400 level	3.0
ELEC 484[2, 5]	1.5
CSC 450, or CSC 460, or ELEC 459[3]	1.5
Two CSC at the 400 level[4]	3.0
Electives	3.0
Total	**15.0**

1. Students who are not exempt from the Undergraduate English requirement, page 29, should choose 1.5 units of first-year English.

2. ELEC 407 and 484 are only offered in the summer term.

3. CSC 450 requires CSC 355 and 360 as prerequisites. CSC 460 requires CSC 355 and 360 and either SENG 321 or 365 as prerequisites. ELEC 459 requires ELEC 407 as prerequisite.

4. One of these courses may be SENG at the 400 level.

5. Or acceptable replacement.

General Degree (BA or BSc - Faculties of Humanities, Science and Social Sciences)

Admission to the General Program

Students intending to complete a General degree in Computer Science will normally register in the faculty of the second area of specialization required in the degree.

Completion of the following set of courses satisfies the requirements for a BA or BSc General Degree in Computer Science as offered by the Faculties of Humanities, Social Sciences and Science. Students wishing to complete a General Program should register in whichever of these three faculties is appropriate based on their second area of specialization.

Year 1

CSC 110, 115
MATH 100 and 101, or 102 and151
MATH 122

Year 2

CSC 212, 225, 230
SENG 265
STAT 252 or 254 or 255 or 260 or ECON 246

Years 3 and 4

A total of 9 additional units of Computer Science courses numbered 300 or higher. Two of these CSC courses can be replaced by SENG courses at a similar level.

MINOR IN COMPUTER SCIENCE

Students in other departments may complete a Minor in Computer Science by completing the Major or Honours requirements of that department, in conjunction with either the Computer Science General Program requirement or by completing the set of courses listed below.

Year 1	
CSC 110, 115	3.0
MATH 100 or 102	1.5
MATH 122	1.5
MATH 151 or any Statistics 200-level (or equivalent) course	1.5

Year 2	
CSC 212, 225, 230	4.5
SENG 265	1.5

Year 3	
3 additional CSC courses numbered 300 or higher (one of these can be replaced by a SENG course at a similar level)	4.5

Note that 200 level and higher courses that fulfill requirements for a Minor cannot form part of the requirements for the Major or Honours degree. Any such course in the Minor program may be replaced by another Computer Science course at the same level or higher.

COMPUTER SCIENCE CO-OPERATIVE EDUCATION PROGRAMS

Please refer to the general description of Co-operative Education at UVic, page 44.

General Regulations

The normal requirements for admission of students to a Computer Science/Mathematics Co-op Program are the completion of CSC 110 and MATH 100 and the following:

1. the completion of at least 4.5 units on their last academic term
2. a minimum grade of B- in any Computer Science courses and a minimum grade of C+ in any Mathematics or Statistics courses taken on their last academic term
3. no grades of F, E or N in courses taken on their last academic term

Students are normally admitted to a program in January after their first term on campus; application for admission should be made before the end of the first term. However, a student may be admitted to a program up to the end of his or her second year. A student will be admitted to a Co-op Program only if there is a satisfactory schedule of academic terms and work terms that will enable the student to complete all co-op requirements.

For students who have completed all of CSC 110 and 115 and MATH 100, 101, 122, the normal requirements for admission to a Computer Science/Mathematics Co-op Program are:

1. the completion of at least 4.5 units on their last academic term.
2. a minimum grade of B- in any of CSC 115, 225, 230 and SENG 265 taken on their last academic term; a minimum grade of C+ in any other Computer Science courses taken on their last academic term; and a minimum grade of C in any Mathematics or Statistics courses taken on their last academic term.
3. no grades of F, E or N in courses taken on their last academic term.

Students registered in a Co-op Program must normally be enrolled in at least 6 units of course work during each academic term. The performance of students will be reviewed after each academic term and each work term. Students who fail to achieve satisfactory standing on an academic term or satisfactory completion of a work term may be required to withdraw from the program. Each work term is recorded on the student's academic record and transcript.

The granting of work term credit by challenge is permitted in all Computer Science Co-op Programs except Health Information Science/Computer Science, and is governed by the regulations on page 45.

Computer Science, Computer Science (Software Engineering Option), Computer Science (Bioinformatics Option), Computer Science/Mathematics and Computer Science/Statistics

Students admitted to one of these programs who wish to participate in Co-op must successfully complete four work terms in order to complete their Co-op degree requirements, and satisfy the course requirements of their specific degree program.

Computer Science (Business Option)

Students admitted to the Major Program in Computer Science (Business Option) are required to take part in the Co-op Education Program. They must successfully complete four work terms. First year students interested in the Business Option should register for ECON 103 and 104 in their fall term, and should contact the Co-op office to arrange for their spring term registration in COM 220.

Computer Science (Mechatronics and Embedded System Option)

Students participating in this program are required to take part in the Co-op Education Program. They must successfully complete four work terms. Interested students should contact the Co-op Office regarding admission during their first term on campus or upon deciding to enter the program.

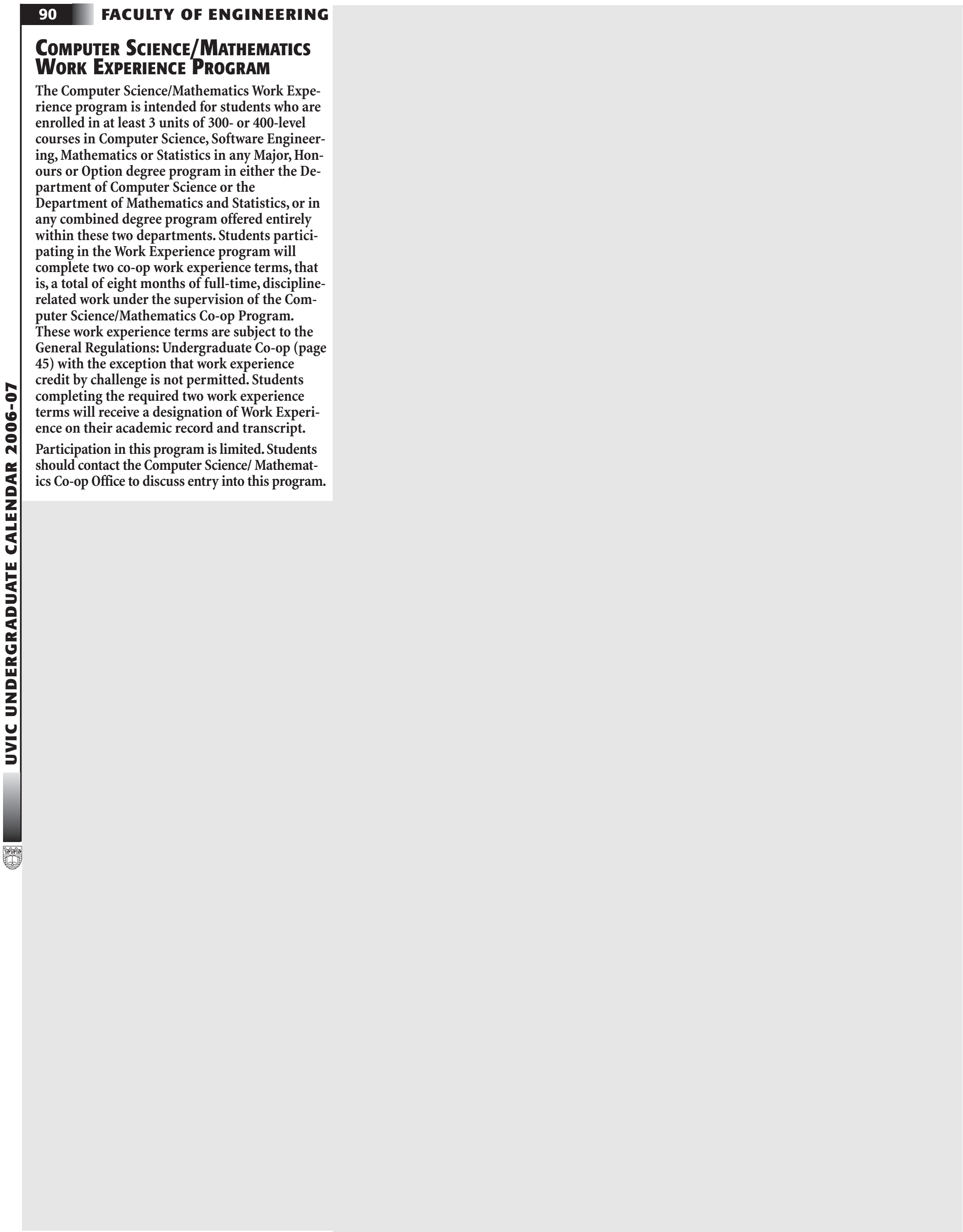
Health Information Science/Computer Science

Students admitted to the Combined Program in Health Information Science and Computer Science are required to take part in the Co-op Education Program. They must successfully complete three work terms with at most two in one department in order to graduate in this program. The granting of work term credit by challenge is not permitted in this program.

Physics/Computer Science, Psychology/Computer Science, Visual Arts/Computer Science, Geography/Computer Science and Music/Computer Science

Students in one of these Combined Programs who wish to participate in Co-op must be admitted by the Co-op program of each department or school involved. They must successfully complete two work terms in each Co-op program in order to complete their Co-op degree requirements.

Computer Science Undergraduate Programs, continued



90 FACULTY OF ENGINEERING

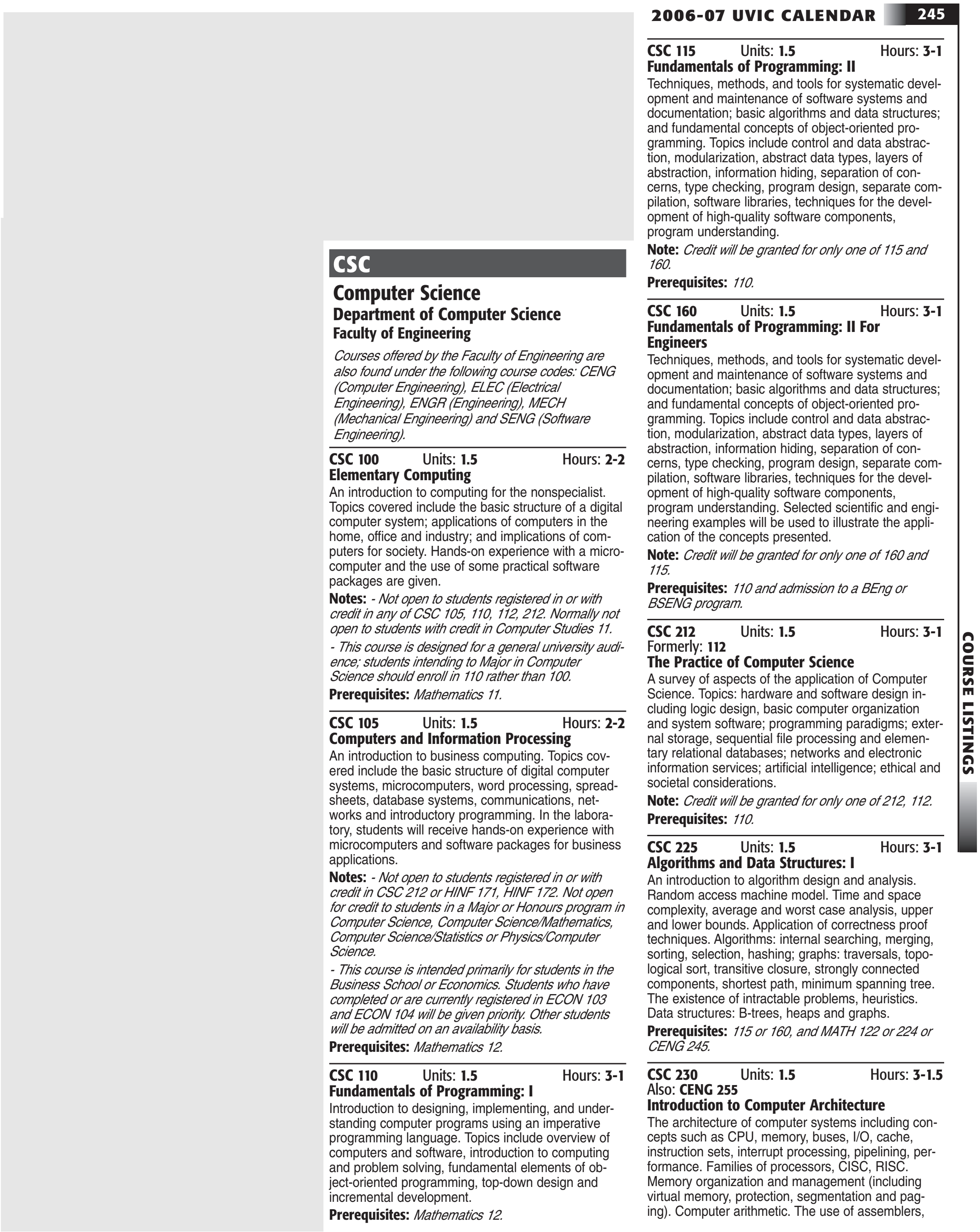
UVIC UNDERGRADUATE CALENDAR 2006-07

Computer Science/Mathematics Work Experience Program

The Computer Science/Mathematics Work Experience program is intended for students who are enrolled in at least 3 units of 300- or 400-level courses in Computer Science, Software Engineering, Mathematics or Statistics in any Major, Honours or Option degree program in either the Department of Computer Science or the Department of Mathematics and Statistics, or in any combined degree program offered entirely within these two departments. Students participating in the Work Experience program will complete two co-op work experience terms, that is, a total of eight months of full-time, discipline-related work under the supervision of the Computer Science/Mathematics Co-op Program. These work experience terms are subject to the General Regulations: Undergraduate Co-op (page 45) with the exception that work experience credit by challenge is not permitted. Students completing the required two work experience terms will receive a designation of Work Experience on their academic record and transcript.

Participation in this program is limited. Students should contact the Computer Science/ Mathematics Co-op Office to discuss entry into this program.

J.2 Computer Science Undergraduate Course Descriptions



CSC

Computer Science

Department of Computer Science

Faculty of Engineering

Courses offered by the Faculty of Engineering are also found under the following course codes: CENG (Computer Engineering), ELEC (Electrical Engineering), ENGR (Engineering), MECH (Mechanical Engineering) and SENG (Software Engineering).

CSC 100 Units: **1.5** Hours: **2-2**
Elementary Computing

An introduction to computing for the nonspecialist. Topics covered include the basic structure of a digital computer system; applications of computers in the home, office and industry; and implications of computers for society. Hands-on experience with a microcomputer and the use of some practical software packages are given.

Notes: *- Not open to students registered in or with credit in any of CSC 105, 110, 112, 212. Normally not open to students with credit in Computer Studies 11.*

- This course is designed for a general university audience; students intending to Major in Computer Science should enroll in 110 rather than 100.

Prerequisites: *Mathematics 11.*

CSC 105 Units: **1.5** Hours: **2-2**
Computers and Information Processing

An introduction to business computing. Topics covered include the basic structure of digital computer systems, microcomputers, word processing, spreadsheets, database systems, communications, networks and introductory programming. In the laboratory, students will receive hands-on experience with microcomputers and software packages for business applications.

Notes: *- Not open to students registered in or with credit in CSC 212 or HINF 171, HINF 172. Not open for credit to students in a Major or Honours program in Computer Science, Computer Science/Mathematics, Computer Science/Statistics or Physics/Computer Science.*

- This course is intended primarily for students in the Business School or Economics. Students who have completed or are currently registered in ECON 103 and ECON 104 will be given priority. Other students will be admitted on an availability basis.

Prerequisites: *Mathematics 12.*

CSC 110 Units: **1.5** Hours: **3-1**
Fundamentals of Programming: I

Introduction to designing, implementing, and understanding computer programs using an imperative programming language. Topics include overview of computers and software, introduction to computing and problem solving, fundamental elements of object-oriented programming, top-down design and incremental development.

Prerequisites: *Mathematics 12.*

CSC 115 Units: **1.5** Hours: **3-1**
Fundamentals of Programming: II

Techniques, methods, and tools for systematic development and maintenance of software systems and documentation; basic algorithms and data structures; and fundamental concepts of object-oriented programming. Topics include control and data abstraction, modularization, abstract data types, layers of abstraction, information hiding, separation of concerns, type checking, program design, separate compilation, software libraries, techniques for the development of high-quality software components, program understanding.

Note: *Credit will be granted for only one of 115 and 160.*

Prerequisites: *110.*

CSC 160 Units: **1.5** Hours: **3-1**
Fundamentals of Programming: II For Engineers

Techniques, methods, and tools for systematic development and maintenance of software systems and documentation; basic algorithms and data structures; and fundamental concepts of object-oriented programming. Topics include control and data abstraction, modularization, abstract data types, layers of abstraction, information hiding, separation of concerns, type checking, program design, separate compilation, software libraries, techniques for the development of high-quality software components, program understanding. Selected scientific and engineering examples will be used to illustrate the application of the concepts presented.

Note: *Credit will be granted for only one of 160 and 115.*

Prerequisites: *110 and admission to a BEng or BSENG program.*

CSC 212 Units: **1.5** Hours: **3-1**
Formerly: **112**
The Practice of Computer Science

A survey of aspects of the application of Computer Science. Topics: hardware and software design including logic design, basic computer organization and system software; programming paradigms; external storage, sequential file processing and elementary relational databases; networks and electronic information services; artificial intelligence; ethical and societal considerations.

Note: *Credit will be granted for only one of 212, 112.*

Prerequisites: *110.*

CSC 225 Units: **1.5** Hours: **3-1**
Algorithms and Data Structures: I

An introduction to algorithm design and analysis. Random access machine model. Time and space complexity, average and worst case analysis, upper and lower bounds. Application of correctness proof techniques. Algorithms: internal searching, merging, sorting, selection, hashing; graphs: traversals, topological sort, transitive closure, strongly connected components, shortest path, minimum spanning tree. The existence of intractable problems, heuristics. Data structures: B-trees, heaps and graphs.

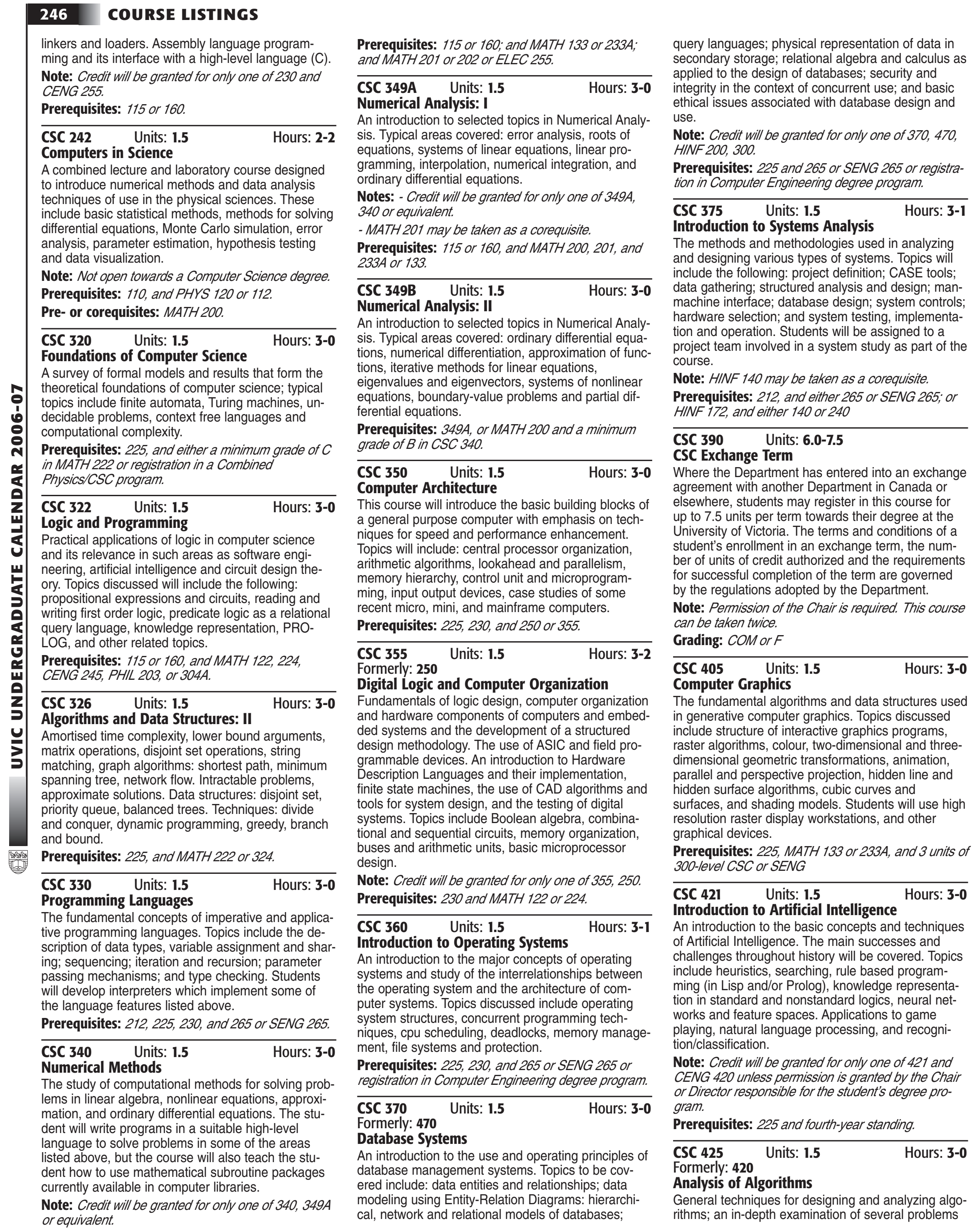
Prerequisites: *115 or 160, and MATH 122 or 224 or CENG 245.*

CSC 230 Units: **1.5** Hours: **3-1.5**
Also: **CENG 255**
Introduction to Computer Architecture

The architecture of computer systems including concepts such as CPU, memory, buses, I/O, cache, instruction sets, interrupt processing, pipelining, performance. Families of processors, CISC, RISC. Memory organization and management (including virtual memory, protection, segmentation and paging). Computer arithmetic. The use of assemblers,

COURSE LISTINGS

Computer Science Undergraduate Course Descriptions, continued



linkers and loaders. Assembly language programming and its interface with a high-level language (C).

Note: *Credit will be granted for only one of 230 and CENG 255.*

Prerequisites: *115 or 160.*

CSC 242 Units: **1.5** Hours: **2-2**
Computers in Science

A combined lecture and laboratory course designed to introduce numerical methods and data analysis techniques of use in the physical sciences. These include basic statistical methods, methods for solving differential equations, Monte Carlo simulation, error analysis, parameter estimation, hypothesis testing and data visualization.

Note: *Not open towards a Computer Science degree.*

Prerequisites: *110, and PHYS 120 or 112.*

Pre- or corequisites: *MATH 200.*

CSC 320 Units: **1.5** Hours: **3-0**
Foundations of Computer Science

A survey of formal models and results that form the theoretical foundations of computer science; typical topics include finite automata, Turing machines, undecidable problems, context free languages and computational complexity.

Prerequisites: *225, and either a minimum grade of C in MATH 222 or registration in a Combined Physics/CSC program.*

CSC 322 Units: **1.5** Hours: **3-0**
Logic and Programming

Practical applications of logic in computer science and its relevance in such areas as software engineering, artificial intelligence and circuit design theory. Topics discussed will include the following: propositional expressions and circuits, reading and writing first order logic, predicate logic as a relational query language, knowledge representation, PROLOG, and other related topics.

Prerequisites: *115 or 160, and MATH 122, 224, CENG 245, PHIL 203, or 304A.*

CSC 326 Units: **1.5** Hours: **3-0**
Algorithms and Data Structures: II

Amortised time complexity, lower bound arguments, matrix operations, disjoint set operations, string matching, graph algorithms: shortest path, minimum spanning tree, network flow. Intractable problems, approximate solutions. Data structures: disjoint set, priority queue, balanced trees. Techniques: divide and conquer, dynamic programming, greedy, branch and bound.

Prerequisites: *225, and MATH 222 or 324.*

CSC 330 Units: **1.5** Hours: **3-0**
Programming Languages

The fundamental concepts of imperative and applicative programming languages. Topics include the description of data types, variable assignment and sharing; sequencing; iteration and recursion; parameter passing mechanisms; and type checking. Students will develop interpreters which implement some of the language features listed above.

Prerequisites: *212, 225, 230, and 265 or SENG 265.*

CSC 340 Units: **1.5** Hours: **3-0**
Numerical Methods

The study of computational methods for solving problems in linear algebra, nonlinear equations, approximation, and ordinary differential equations. The student will write programs in a suitable high-level language to solve problems in some of the areas listed above, but the course will also teach the student how to use mathematical subroutine packages currently available in computer libraries.

Note: *Credit will be granted for only one of 340, 349A or equivalent.*

Prerequisites: *115 or 160; and MATH 133 or 233A; and MATH 201 or 202 or ELEC 255.*

CSC 349A Units: **1.5** Hours: **3-0**
Numerical Analysis: I

An introduction to selected topics in Numerical Analysis. Typical areas covered: error analysis, roots of equations, systems of linear equations, linear programming, interpolation, numerical integration, and ordinary differential equations.

Notes: *- Credit will be granted for only one of 349A, 340 or equivalent.*

- MATH 201 may be taken as a corequisite.

Prerequisites: *115 or 160, and MATH 200, 201, and 233A or 133.*

CSC 349B Units: **1.5** Hours: **3-0**
Numerical Analysis: II

An introduction to selected topics in Numerical Analysis. Typical areas covered: ordinary differential equations, numerical differentiation, approximation of functions, iterative methods for linear equations, eigenvalues and eigenvectors, systems of nonlinear equations, boundary-value problems and partial differential equations.

Prerequisites: *349A, or MATH 200 and a minimum grade of B in CSC 340.*

CSC 350 Units: **1.5** Hours: **3-0**
Computer Architecture

This course will introduce the basic building blocks of a general purpose computer with emphasis on techniques for speed and performance enhancement. Topics will include: central processor organization, arithmetic algorithms, lookahead and parallelism, memory hierarchy, control unit and microprogramming, input output devices, case studies of some recent micro, mini, and mainframe computers.

Prerequisites: *225, 230, and 250 or 355.*

CSC 355 Units: **1.5** Hours: **3-2**
Formerly: **250**
Digital Logic and Computer Organization

Fundamentals of logic design, computer organization and hardware components of computers and embedded systems and the development of a structured design methodology. The use of ASIC and field programmable devices. An introduction to Hardware Description Languages and their implementation, finite state machines, the use of CAD algorithms and tools for system design, and the testing of digital systems. Topics include Boolean algebra, combinational and sequential circuits, memory organization, buses and arithmetic units, basic microprocessor design.

Note: *Credit will be granted for only one of 355, 250.*

Prerequisites: *230 and MATH 122 or 224.*

CSC 360 Units: **1.5** Hours: **3-1**
Introduction to Operating Systems

An introduction to the major concepts of operating systems and study of the interrelationships between the operating system and the architecture of computer systems. Topics discussed include operating system structures, concurrent programming techniques, cpu scheduling, deadlocks, memory management, file systems and protection.

Prerequisites: *225, 230, and 265 or SENG 265 or registration in Computer Engineering degree program.*

CSC 370 Units: **1.5** Hours: **3-0**
Formerly: **470**
Database Systems

An introduction to the use and operating principles of database management systems. Topics to be covered include: data entities and relationships; data modeling using Entity-Relation Diagrams: hierarchical, network and relational models of databases; query languages; physical representation of data in secondary storage; relational algebra and calculus as applied to the design of databases; security and integrity in the context of concurrent use; and basic ethical issues associated with database design and use.

Note: *Credit will be granted for only one of 370, 470, HINF 200, 300.*

Prerequisites: *225 and 265 or SENG 265 or registration in Computer Engineering degree program.*

CSC 375 Units: **1.5** Hours: **3-1**
Introduction to Systems Analysis

The methods and methodologies used in analyzing and designing various types of systems. Topics will include the following: project definition; CASE tools; data gathering; structured analysis and design; man-machine interface; database design; system controls; hardware selection; and system testing, implementation and operation. Students will be assigned to a project team involved in a system study as part of the course.

Note: *HINF 140 may be taken as a corequisite.*

Prerequisites: *212, and either 265 or SENG 265; or HINF 172, and either 140 or 240*

CSC 390 Units: **6.0-7.5**
CSC Exchange Term

Where the Department has entered into an exchange agreement with another Department in Canada or elsewhere, students may register in this course for up to 7.5 units per term towards their degree at the University of Victoria. The terms and conditions of a student's enrollment in an exchange term, the number of units of credit authorized and the requirements for successful completion of the term are governed by the regulations adopted by the Department.

Note: *Permission of the Chair is required. This course can be taken twice.*

Grading: *COM or F*

CSC 405 Units: **1.5** Hours: **3-0**
Computer Graphics

The fundamental algorithms and data structures used in generative computer graphics. Topics discussed include structure of interactive graphics programs, raster algorithms, colour, two-dimensional and three-dimensional geometric transformations, animation, parallel and perspective projection, hidden line and hidden surface algorithms, cubic curves and surfaces, and shading models. Students will use high resolution raster display workstations, and other graphical devices.

Prerequisites: *225, MATH 133 or 233A, and 3 units of 300-level CSC or SENG*

CSC 421 Units: **1.5** Hours: **3-0**
Introduction to Artificial Intelligence

An introduction to the basic concepts and techniques of Artificial Intelligence. The main successes and challenges throughout history will be covered. Topics include heuristics, searching, rule based programming (in Lisp and/or Prolog), knowledge representation in standard and nonstandard logics, neural networks and feature spaces. Applications to game playing, natural language processing, and recognition/classification.

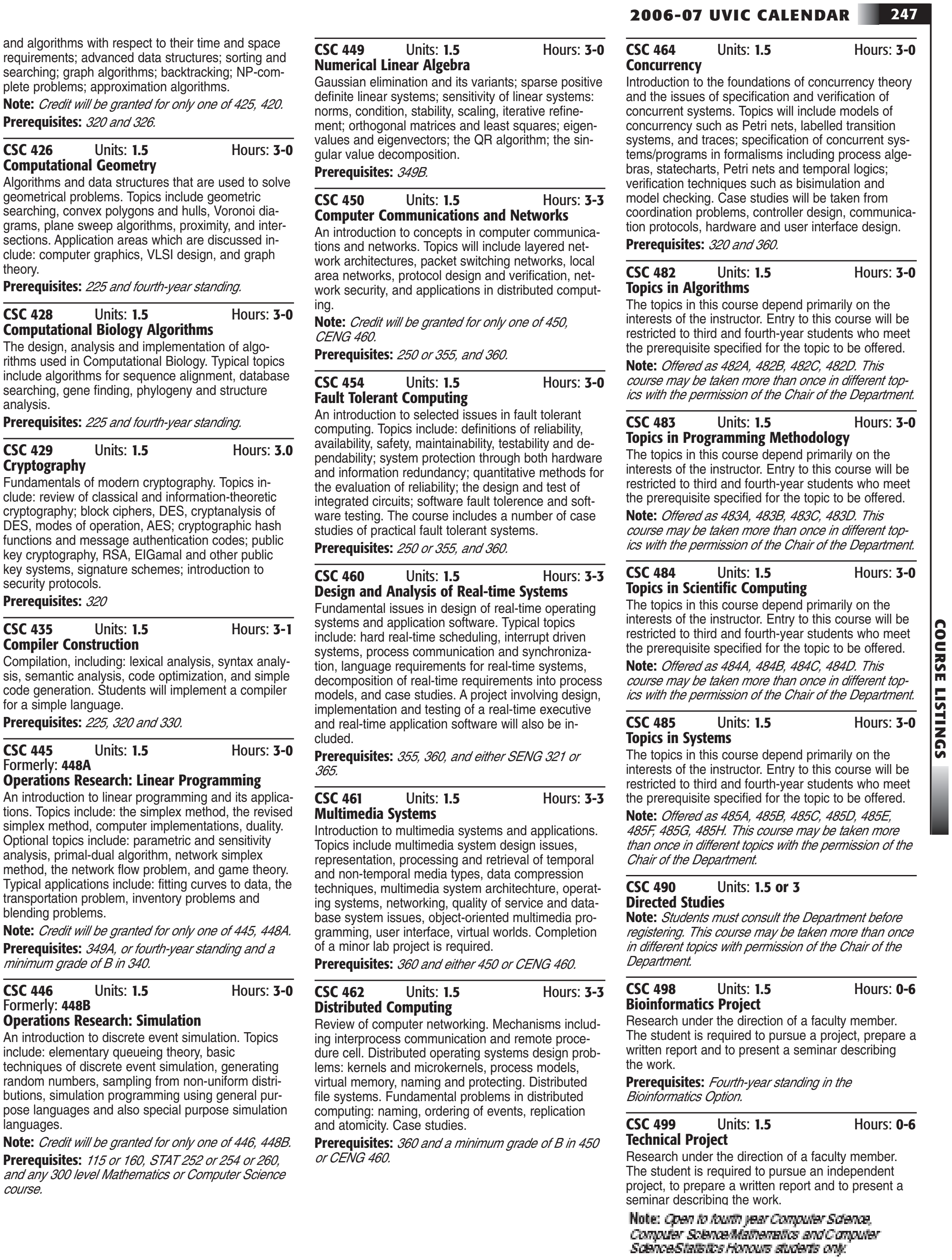
Note: *Credit will be granted for only one of 421 and CENG 420 unless permission is granted by the Chair or Director responsible for the student's degree program.*

Prerequisites: *225 and fourth-year standing.*

CSC 425 Units: **1.5** Hours: **3-0**
Formerly: **420**
Analysis of Algorithms

General techniques for designing and analyzing algorithms; an in-depth examination of several problems

Computer Science Undergraduate Course Descriptions, continued



and algorithms with respect to their time and space requirements; advanced data structures; sorting and searching; graph algorithms; backtracking; NP-complete problems; approximation algorithms.

Note: *Credit will be granted for only one of 425, 420.*

Prerequisites: *320 and 326.*

CSC 426 Units: **1.5** Hours: **3-0**

Computational Geometry

Algorithms and data structures that are used to solve geometrical problems. Topics include geometric searching, convex polygons and hulls, Voronoi diagrams, plane sweep algorithms, proximity, and intersections. Application areas which are discussed include: computer graphics, VLSI design, and graph theory.

Prerequisites: *225 and fourth-year standing.*

CSC 428 Units: **1.5** Hours: **3-0**

Computational Biology Algorithms

The design, analysis and implementation of algorithms used in Computational Biology. Typical topics include algorithms for sequence alignment, database searching, gene finding, phylogeny and structure analysis.

Prerequisites: *225 and fourth-year standing.*

CSC 429 Units: **1.5** Hours: **3.0**

Cryptography

Fundamentals of modern cryptography. Topics include: review of classical and information-theoretic cryptography; block ciphers, DES, cryptanalysis of DES, modes of operation, AES; cryptographic hash functions and message authentication codes; public key cryptography, RSA, ElGamal and other public key systems, signature schemes; introduction to security protocols.

Prerequisites: *320*

CSC 435 Units: **1.5** Hours: **3-1**

Compiler Construction

Compilation, including: lexical analysis, syntax analysis, semantic analysis, code optimization, and simple code generation. Students will implement a compiler for a simple language.

Prerequisites: *225, 320 and 330.*

CSC 445 Units: **1.5** Hours: **3-0**

Formerly: **448A**

Operations Research: Linear Programming

An introduction to linear programming and its applications. Topics include: the simplex method, the revised simplex method, computer implementations, duality. Optional topics include: parametric and sensitivity analysis, primal-dual algorithm, network simplex method, the network flow problem, and game theory. Typical applications include: fitting curves to data, the transportation problem, inventory problems and blending problems.

Note: *Credit will be granted for only one of 445, 448A.*

Prerequisites: *349A, or fourth-year standing and a minimum grade of B in 340.*

CSC 446 Units: **1.5** Hours: **3-0**

Formerly: **448B**

Operations Research: Simulation

An introduction to discrete event simulation. Topics include: elementary queueing theory, basic techniques of discrete event simulation, generating random numbers, sampling from non-uniform distributions, simulation programming using general purpose languages and also special purpose simulation languages.

Note: *Credit will be granted for only one of 446, 448B.*

Prerequisites: *115 or 160, STAT 252 or 254 or 260, and any 300 level Mathematics or Computer Science course.*

CSC 449 Units: **1.5** Hours: **3-0**

Numerical Linear Algebra

Gaussian elimination and its variants; sparse positive definite linear systems; sensitivity of linear systems: norms, condition, stability, scaling, iterative refinement; orthogonal matrices and least squares; eigenvalues and eigenvectors; the QR algorithm; the singular value decomposition.

Prerequisites: *349B.*

CSC 450 Units: **1.5** Hours: **3-3**

Computer Communications and Networks

An introduction to concepts in computer communications and networks. Topics will include layered network architectures, packet switching networks, local area networks, protocol design and verification, network security, and applications in distributed computing.

Note: *Credit will be granted for only one of 450, CENG 460.*

Prerequisites: *250 or 355, and 360.*

CSC 454 Units: **1.5** Hours: **3-0**

Fault Tolerant Computing

An introduction to selected issues in fault tolerant computing. Topics include: definitions of reliability, availability, safety, maintainability, testability and dependability; system protection through both hardware and information redundancy; quantitative methods for the evaluation of reliability; the design and test of integrated circuits; software fault tolerence and software testing. The course includes a number of case studies of practical fault tolerant systems.

Prerequisites: *250 or 355, and 360.*

CSC 460 Units: **1.5** Hours: **3-3**

Design and Analysis of Real-time Systems

Fundamental issues in design of real-time operating systems and application software. Typical topics include: hard real-time scheduling, interrupt driven systems, process communication and synchronization, language requirements for real-time systems, decomposition of real-time requirements into process models, and case studies. A project involving design, implementation and testing of a real-time executive and real-time application software will also be included.

Prerequisites: *355, 360, and either SENG 321 or 365.*

CSC 461 Units: **1.5** Hours: **3-3**

Multimedia Systems

Introduction to multimedia systems and applications. Topics include multimedia system design issues, representation, processing and retrieval of temporal and non-temporal media types, data compression techniques, multimedia system archichtecture, operating systems, networking, quality of service and database system issues, object-oriented multimedia programming, user interface, virtual worlds. Completion of a minor lab project is required.

Prerequisites: *360 and either 450 or CENG 460.*

CSC 462 Units: **1.5** Hours: **3-3**

Distributed Computing

Review of computer networking. Mechanisms including interprocess communication and remote procedure cell. Distributed operating systems design problems: kernels and microkernels, process models, virtual memory, naming and protecting. Distributed file systems. Fundamental problems in distributed computing: naming, ordering of events, replication and atomicity. Case studies.

Prerequisites: *360 and a minimum grade of B in 450 or CENG 460.*

CSC 464 Units: **1.5** Hours: **3-0**

Concurrency

Introduction to the foundations of concurrency theory and the issues of specification and verification of concurrent systems. Topics will include models of concurrency such as Petri nets, labelled transition systems, and traces; specification of concurrent systems/programs in formalisms including process algebras, statecharts, Petri nets and temporal logics; verification techniques such as bisimulation and model checking. Case studies will be taken from coordination problems, controller design, communication protocols, hardware and user interface design.

Prerequisites: *320 and 360.*

CSC 482 Units: **1.5** Hours: **3-0**

Topics in Algorithms

The topics in this course depend primarily on the interests of the instructor. Entry to this course will be restricted to third and fourth-year students who meet the prerequisite specified for the topic to be offered.

Note: *Offered as 482A, 482B, 482C, 482D. This course may be taken more than once in different topics with the permission of the Chair of the Department.*

CSC 483 Units: **1.5** Hours: **3-0**

Topics in Programming Methodology

The topics in this course depend primarily on the interests of the instructor. Entry to this course will be restricted to third and fourth-year students who meet the prerequisite specified for the topic to be offered.

Note: *Offered as 483A, 483B, 483C, 483D. This course may be taken more than once in different topics with the permission of the Chair of the Department.*

CSC 484 Units: **1.5** Hours: **3-0**

Topics in Scientific Computing

The topics in this course depend primarily on the interests of the instructor. Entry to this course will be restricted to third and fourth-year students who meet the prerequisite specified for the topic to be offered.

Note: *Offered as 484A, 484B, 484C, 484D. This course may be taken more than once in different topics with the permission of the Chair of the Department.*

CSC 485 Units: **1.5** Hours: **3-0**

Topics in Systems

The topics in this course depend primarily on the interests of the instructor. Entry to this course will be restricted to third and fourth-year students who meet the prerequisite specified for the topic to be offered.

Note: *Offered as 485A, 485B, 485C, 485D, 485E, 485F, 485G, 485H. This course may be taken more than once in different topics with the permission of the Chair of the Department.*

CSC 490 Units: **1.5 or 3**

Directed Studies

Note: *Students must consult the Department before registering. This course may be taken more than once in different topics with permission of the Chair of the Department.*

CSC 498 Units: **1.5** Hours: **0-6**

Bioinformatics Project

Research under the direction of a faculty member. The student is required to pursue a project, prepare a written report and to present a seminar describing the work.

Prerequisites: *Fourth-year standing in the Bioinformatics Option.*

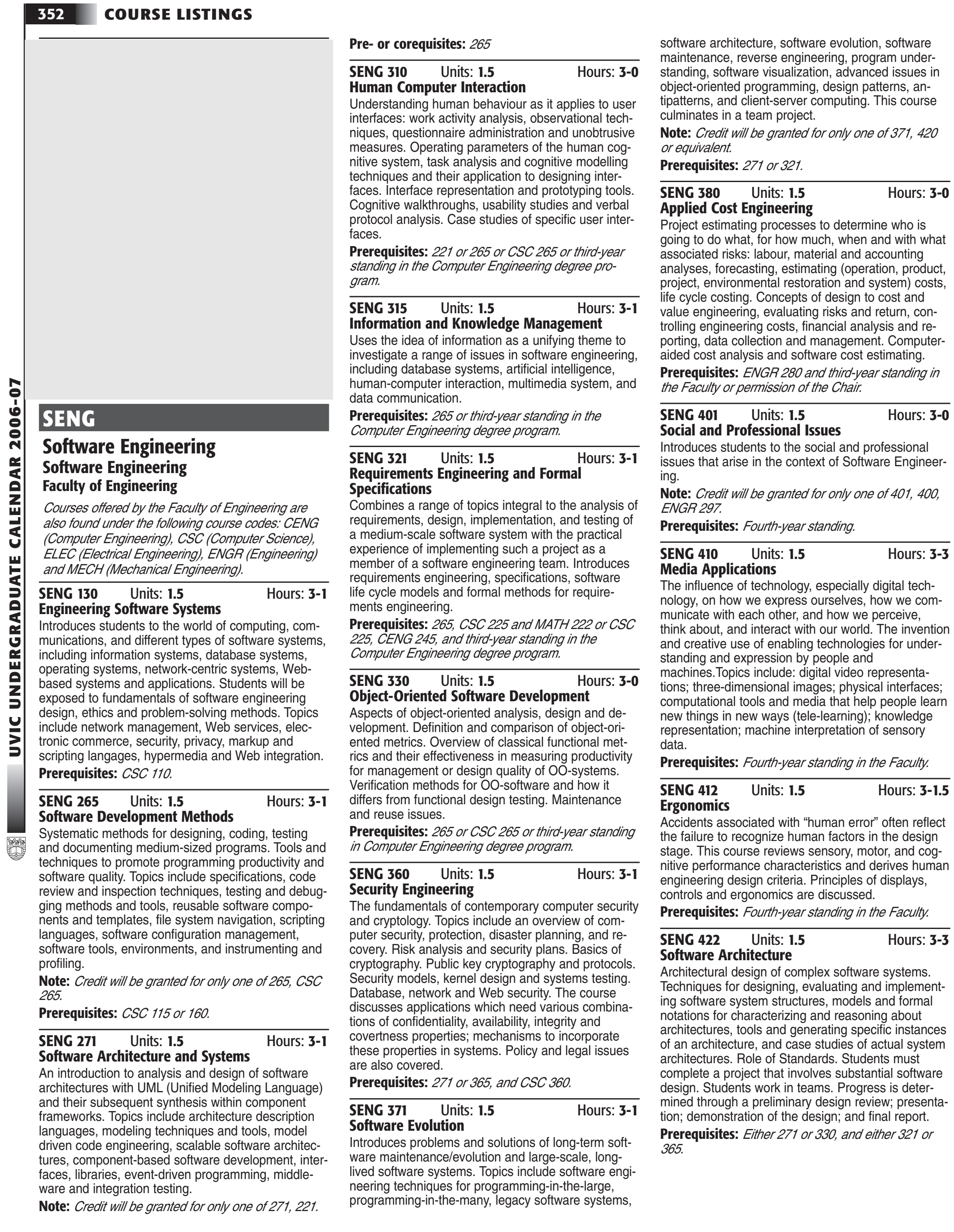
CSC 499 Units: **1.5** Hours: **0-6**

Technical Project

Research under the direction of a faculty member. The student is required to pursue an independent project, to prepare a written report and to present a seminar describing the work.

Note: *Open to fourth year Computer Science, Computer Science/Mathematics and Computer Science/Statistics Honours students only.*

J.3 SENG Undergraduate Course Descriptions



SENG

Software Engineering

Software Engineering
Faculty of Engineering

Courses offered by the Faculty of Engineering are also found under the following course codes: CENG (Computer Engineering), CSC (Computer Science), ELEC (Electrical Engineering), ENGR (Engineering) and MECH (Mechanical Engineering).

SENG 130 Units: **1.5** Hours: **3-1**
Engineering Software Systems
Introduces students to the world of computing, communications, and different types of software systems, including information systems, database systems, operating systems, network-centric systems, Web-based systems and applications. Students will be exposed to fundamentals of software engineering design, ethics and problem-solving methods. Topics include network management, Web services, electronic commerce, security, privacy, markup and scripting langages, hypermedia and Web integration.

Prerequisites: *CSC 110.*

SENG 265 Units: **1.5** Hours: **3-1**
Software Development Methods
Systematic methods for designing, coding, testing and documenting medium-sized programs. Tools and techniques to promote programming productivity and software quality. Topics include specifications, code review and inspection techniques, testing and debugging methods and tools, reusable software components and templates, file system navigation, scripting languages, software configuration management, software tools, environments, and instrumenting and profiling.

Note: *Credit will be granted for only one of 265, CSC 265.*

Prerequisites: *CSC 115 or 160.*

SENG 271 Units: **1.5** Hours: **3-1**
Software Architecture and Systems
An introduction to analysis and design of software architectures with UML (Unified Modeling Language) and their subsequent synthesis within component frameworks. Topics include architecture description languages, modeling techniques and tools, model driven code engineering, scalable software architectures, component-based software development, interfaces, libraries, event-driven programming, middleware and integration testing.

Note: *Credit will be granted for only one of 271, 221.*

Pre- or corequisites: *265*

SENG 310 Units: **1.5** Hours: **3-0**
Human Computer Interaction
Understanding human behaviour as it applies to user interfaces: work activity analysis, observational techniques, questionnaire administration and unobtrusive measures. Operating parameters of the human cognitive system, task analysis and cognitive modelling techniques and their application to designing interfaces. Interface representation and prototyping tools. Cognitive walkthroughs, usability studies and verbal protocol analysis. Case studies of specific user interfaces.

Prerequisites: *221 or 265 or CSC 265 or third-year standing in the Computer Engineering degree program.*

SENG 315 Units: **1.5** Hours: **3-1**
Information and Knowledge Management
Uses the idea of information as a unifying theme to investigate a range of issues in software engineering, including database systems, artificial intelligence, human-computer interaction, multimedia system, and data communication.

Prerequisites: *265 or third-year standing in the Computer Engineering degree program.*

SENG 321 Units: **1.5** Hours: **3-1**
Requirements Engineering and Formal Specifications
Combines a range of topics integral to the analysis of requirements, design, implementation, and testing of a medium-scale software system with the practical experience of implementing such a project as a member of a software engineering team. Introduces requirements engineering, specifications, software life cycle models and formal methods for requirements engineering.

Prerequisites: *265, CSC 225 and MATH 222 or CSC 225, CENG 245, and third-year standing in the Computer Engineering degree program.*

SENG 330 Units: **1.5** Hours: **3-0**
Object-Oriented Software Development
Aspects of object-oriented analysis, design and development. Definition and comparison of object-oriented metrics. Overview of classical functional metrics and their effectiveness in measuring productivity for management or design quality of OO-systems. Verification methods for OO-software and how it differs from functional design testing. Maintenance and reuse issues.

Prerequisites: *265 or CSC 265 or third-year standing in Computer Engineering degree program.*

SENG 360 Units: **1.5** Hours: **3-1**
Security Engineering
The fundamentals of contemporary computer security and cryptology. Topics include an overview of computer security, protection, disaster planning, and recovery. Risk analysis and security plans. Basics of cryptography. Public key cryptography and protocols. Security models, kernel design and systems testing. Database, network and Web security. The course discusses applications which need various combinations of confidentiality, availability, integrity and covertness properties; mechanisms to incorporate these properties in systems. Policy and legal issues are also covered.

Prerequisites: *271 or 365, and CSC 360.*

SENG 371 Units: **1.5** Hours: **3-1**
Software Evolution
Introduces problems and solutions of long-term software maintenance/evolution and large-scale, long-lived software systems. Topics include software engineering techniques for programming-in-the-large, programming-in-the-many, legacy software systems, software architecture, software evolution, software maintenance, reverse engineering, program understanding, software visualization, advanced issues in object-oriented programming, design patterns, antipatterns, and client-server computing. This course culminates in a team project.

Note: *Credit will be granted for only one of 371, 420 or equivalent.*

Prerequisites: *271 or 321.*

SENG 380 Units: **1.5** Hours: **3-0**
Applied Cost Engineering
Project estimating processes to determine who is going to do what, for how much, when and with what associated risks: labour, material and accounting analyses, forecasting, estimating (operation, product, project, environmental restoration and system) costs, life cycle costing. Concepts of design to cost and value engineering, evaluating risks and return, controlling engineering costs, financial analysis and reporting, data collection and management. Computer-aided cost analysis and software cost estimating.

Prerequisites: *ENGR 280 and third-year standing in the Faculty or permission of the Chair.*

SENG 401 Units: **1.5** Hours: **3-0**
Social and Professional Issues
Introduces students to the social and professional issues that arise in the context of Software Engineering.

Note: *Credit will be granted for only one of 401, 400, ENGR 297.*

Prerequisites: *Fourth-year standing.*

SENG 410 Units: **1.5** Hours: **3-3**
Media Applications
The influence of technology, especially digital technology, on how we express ourselves, how we communicate with each other, and how we perceive, think about, and interact with our world. The invention and creative use of enabling technologies for understanding and expression by people and machines.Topics include: digital video representations; three-dimensional images; physical interfaces; computational tools and media that help people learn new things in new ways (tele-learning); knowledge representation; machine interpretation of sensory data.

Prerequisites: *Fourth-year standing in the Faculty.*

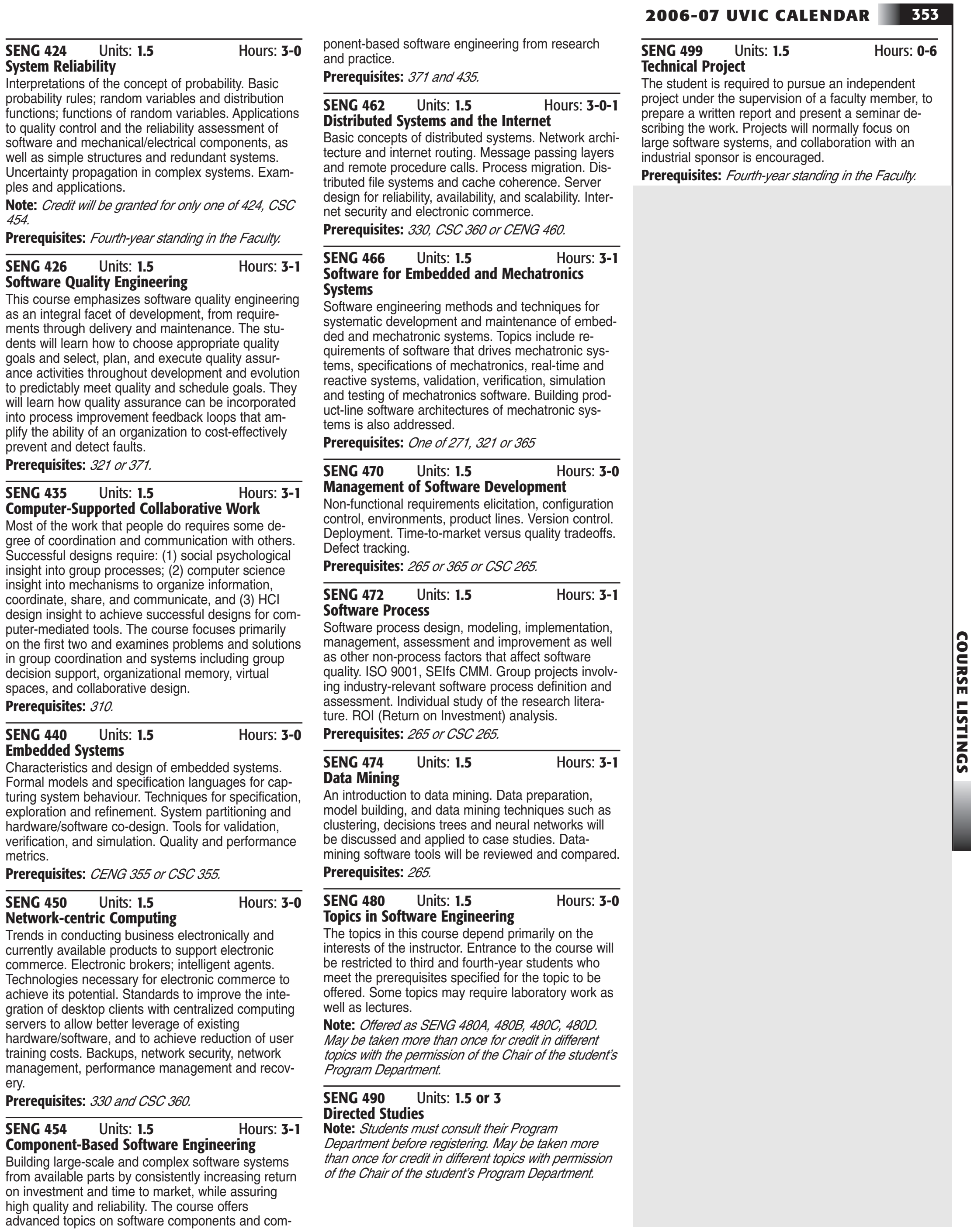
SENG 412 Units: **1.5** Hours: **3-1.5**
Ergonomics
Accidents associated with "human error" often reflect the failure to recognize human factors in the design stage. This course reviews sensory, motor, and cognitive performance characteristics and derives human engineering design criteria. Principles of displays, controls and ergonomics are discussed.

Prerequisites: *Fourth-year standing in the Faculty.*

SENG 422 Units: **1.5** Hours: **3-3**
Software Architecture
Architectural design of complex software systems. Techniques for designing, evaluating and implementing software system structures, models and formal notations for characterizing and reasoning about architectures, tools and generating specific instances of an architecture, and case studies of actual system architectures. Role of Standards. Students must complete a project that involves substantial software design. Students work in teams. Progress is determined through a preliminary design review; presentation; demonstration of the design; and final report.

Prerequisites: *Either 271 or 330, and either 321 or 365.*

SENG Undergraduate Course Descriptions, continued



SENG 424 Units: **1.5** Hours: **3-0**
System Reliability
Interpretations of the concept of probability. Basic probability rules; random variables and distribution functions; functions of random variables. Applications to quality control and the reliability assessment of software and mechanical/electrical components, as well as simple structures and redundant systems. Uncertainty propagation in complex systems. Examples and applications.

Note: *Credit will be granted for only one of 424, CSC 454.*

Prerequisites: *Fourth-year standing in the Faculty.*

SENG 426 Units: **1.5** Hours: **3-1**
Software Quality Engineering
This course emphasizes software quality engineering as an integral facet of development, from requirements through delivery and maintenance. The students will learn how to choose appropriate quality goals and select, plan, and execute quality assurance activities throughout development and evolution to predictably meet quality and schedule goals. They will learn how quality assurance can be incorporated into process improvement feedback loops that amplify the ability of an organization to cost-effectively prevent and detect faults.

Prerequisites: *321 or 371.*

SENG 435 Units: **1.5** Hours: **3-1**
Computer-Supported Collaborative Work
Most of the work that people do requires some degree of coordination and communication with others. Successful designs require: (1) social psychological insight into group processes; (2) computer science insight into mechanisms to organize information, coordinate, share, and communicate, and (3) HCI design insight to achieve successful designs for computer-mediated tools. The course focuses primarily on the first two and examines problems and solutions in group coordination and systems including group decision support, organizational memory, virtual spaces, and collaborative design.

Prerequisites: *310.*

SENG 440 Units: **1.5** Hours: **3-0**
Embedded Systems
Characteristics and design of embedded systems. Formal models and specification languages for capturing system behaviour. Techniques for specification, exploration and refinement. System partitioning and hardware/software co-design. Tools for validation, verification, and simulation. Quality and performance metrics.

Prerequisites: *CENG 355 or CSC 355.*

SENG 450 Units: **1.5** Hours: **3-0**
Network-centric Computing
Trends in conducting business electronically and currently available products to support electronic commerce. Electronic brokers; intelligent agents. Technologies necessary for electronic commerce to achieve its potential. Standards to improve the integration of desktop clients with centralized computing servers to allow better leverage of existing hardware/software, and to achieve reduction of user training costs. Backups, network security, network management, performance management and recovery.

Prerequisites: *330 and CSC 360.*

SENG 454 Units: **1.5** Hours: **3-1**
Component-Based Software Engineering
Building large-scale and complex software systems from available parts by consistently increasing return on investment and time to market, while assuring high quality and reliability. The course offers advanced topics on software components and component-based software engineering from research and practice.

Prerequisites: *371 and 435.*

SENG 462 Units: **1.5** Hours: **3-0-1**
Distributed Systems and the Internet
Basic concepts of distributed systems. Network architecture and internet routing. Message passing layers and remote procedure calls. Process migration. Distributed file systems and cache coherence. Server design for reliability, availability, and scalability. Internet security and electronic commerce.

Prerequisites: *330, CSC 360 or CENG 460.*

SENG 466 Units: **1.5** Hours: **3-1**
Software for Embedded and Mechatronics Systems
Software engineering methods and techniques for systematic development and maintenance of embedded and mechatronic systems. Topics include requirements of software that drives mechatronic systems, specifications of mechatronics, real-time and reactive systems, validation, verification, simulation and testing of mechatronics software. Building product-line software architectures of mechatronic systems is also addressed.

Prerequisites: *One of 271, 321 or 365*

SENG 470 Units: **1.5** Hours: **3-0**
Management of Software Development
Non-functional requirements elicitation, configuration control, environments, product lines. Version control. Deployment. Time-to-market versus quality tradeoffs. Defect tracking.

Prerequisites: *265 or 365 or CSC 265.*

SENG 472 Units: **1.5** Hours: **3-1**
Software Process
Software process design, modeling, implementation, management, assessment and improvement as well as other non-process factors that affect software quality. ISO 9001, SEIfs CMM. Group projects involving industry-relevant software process definition and assessment. Individual study of the research literature. ROI (Return on Investment) analysis.

Prerequisites: *265 or CSC 265.*

SENG 474 Units: **1.5** Hours: **3-1**
Data Mining
An introduction to data mining. Data preparation, model building, and data mining techniques such as clustering, decisions trees and neural networks will be discussed and applied to case studies. Datamining software tools will be reviewed and compared.

Prerequisites: *265.*

SENG 480 Units: **1.5** Hours: **3-0**
Topics in Software Engineering
The topics in this course depend primarily on the interests of the instructor. Entrance to the course will be restricted to third and fourth-year students who meet the prerequisites specified for the topic to be offered. Some topics may require laboratory work as well as lectures.

Note: *Offered as SENG 480A, 480B, 480C, 480D. May be taken more than once for credit in different topics with the permission of the Chair of the student's Program Department.*

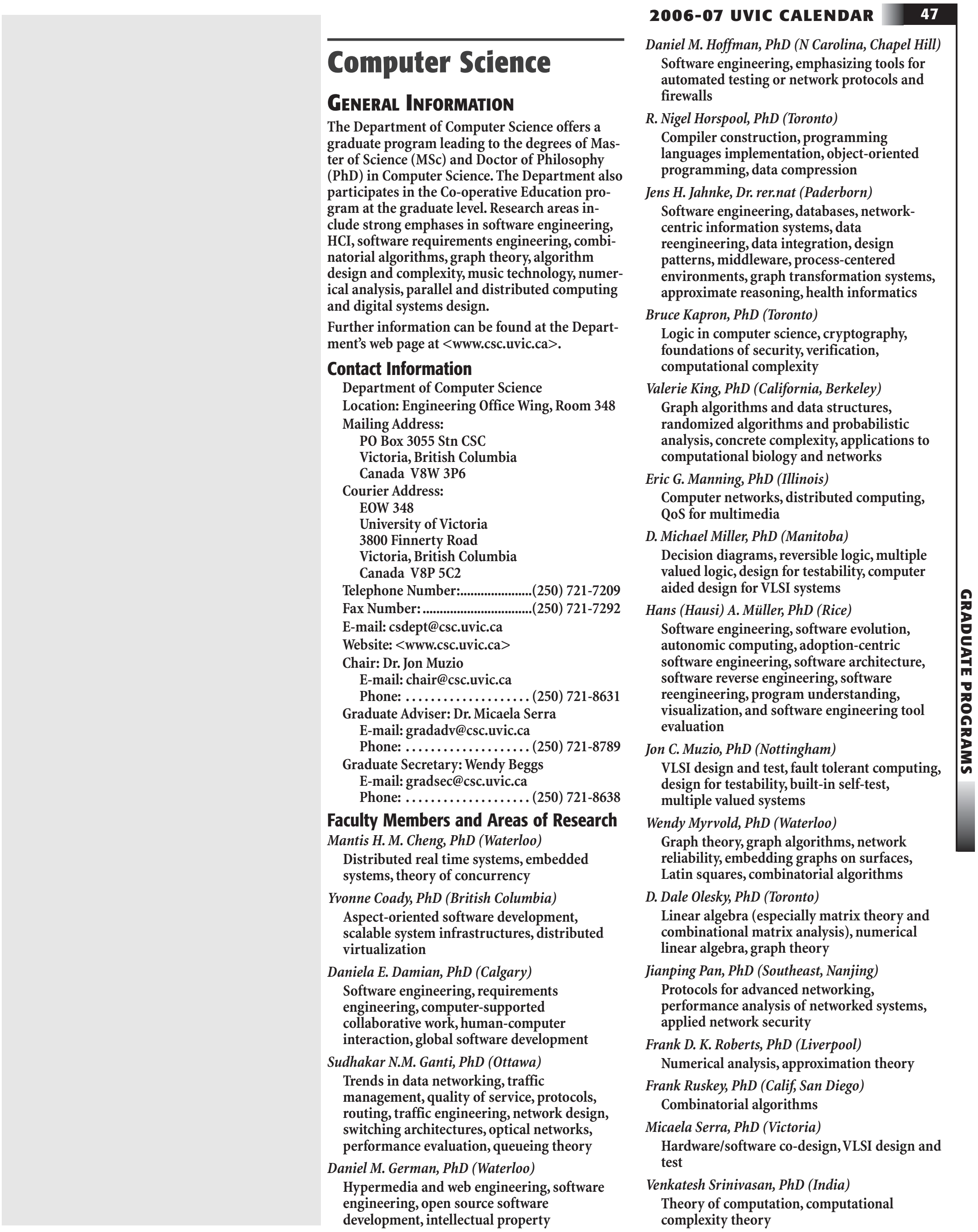
SENG 490 Units: **1.5 or 3**
Directed Studies
Note: *Students must consult their Program Department before registering. May be taken more than once for credit in different topics with permission of the Chair of the student's Program Department.*

SENG 499 Units: **1.5** Hours: **0-6**
Technical Project
The student is required to pursue an independent project under the supervision of a faculty member, to prepare a written report and present a seminar describing the work. Projects will normally focus on large software systems, and collaboration with an industrial sponsor is encouraged.

Prerequisites: *Fourth-year standing in the Faculty.*

COURSE LISTINGS

J.4 Graduate Programs



Computer Science

General Information

The Department of Computer Science offers a graduate program leading to the degrees of Master of Science (MSc) and Doctor of Philosophy (PhD) in Computer Science. The Department also participates in the Co-operative Education program at the graduate level. Research areas include strong emphases in software engineering, HCI, software requirements engineering, combinatorial algorithms, graph theory, algorithm design and complexity, music technology, numerical analysis, parallel and distributed computing and digital systems design.

Further information can be found at the Department's web page at <www.csc.uvic.ca>.

Contact Information

Department of Computer Science
Location: Engineering Office Wing, Room 348
Mailing Address:
PO Box 3055 Stn CSC
Victoria, British Columbia
Canada V8W 3P6
Courier Address:
EOW 348
University of Victoria
3800 Finnerty Road
Victoria, British Columbia
Canada V8P 5C2
Telephone Number:....................(250) 721-7209
Fax Number:...............................(250) 721-7292
E-mail: csdept@csc.uvic.ca
Website: <www.csc.uvic.ca>
Chair: Dr. Jon Muzio
E-mail: chair@csc.uvic.ca
Phone: .... (250) 721-8631
Graduate Adviser: Dr. Micaela Serra
E-mail: gradadv@csc.uvic.ca
Phone: .... (250) 721-8789
Graduate Secretary: Wendy Beggs
E-mail: gradsec@csc.uvic.ca
Phone: .... (250) 721-8638

Faculty Members and Areas of Research

Mantis H. M. Cheng, PhD (Waterloo)
Distributed real time systems, embedded systems, theory of concurrency

Yvonne Coady, PhD (British Columbia)
Aspect-oriented software development, scalable system infrastructures, distributed virtualization

Daniela E. Damian, PhD (Calgary)
Software engineering, requirements engineering, computer-supported collaborative work, human-computer interaction, global software development

Sudhakar N.M. Ganti, PhD (Ottawa)
Trends in data networking, traffic management, quality of service, protocols, routing, traffic engineering, network design, switching architectures, optical networks, performance evaluation, queueing theory

Daniel M. German, PhD (Waterloo)
Hypermedia and web engineering, software engineering, open source software development, intellectual property

Daniel M. Hoffman, PhD (N Carolina, Chapel Hill)
Software engineering, emphasizing tools for automated testing or network protocols and firewalls

R. Nigel Horspool, PhD (Toronto)
Compiler construction, programming languages implementation, object-oriented programming, data compression

Jens H. Jahnke, Dr. rer.nat (Paderborn)
Software engineering, databases, network-centric information systems, data reengineering, data integration, design patterns, middleware, process-centered environments, graph transformation systems, approximate reasoning, health informatics

Bruce Kapron, PhD (Toronto)
Logic in computer science, cryptography, foundations of security, verification, computational complexity

Valerie King, PhD (California, Berkeley)
Graph algorithms and data structures, randomized algorithms and probabilistic analysis, concrete complexity, applications to computational biology and networks

Eric G. Manning, PhD (Illinois)
Computer networks, distributed computing, QoS for multimedia

D. Michael Miller, PhD (Manitoba)
Decision diagrams, reversible logic, multiple valued logic, design for testability, computer aided design for VLSI systems

Hans (Hausi) A. Müller, PhD (Rice)
Software engineering, software evolution, autonomic computing, adoption-centric software engineering, software architecture, software reverse engineering, software reengineering, program understanding, visualization, and software engineering tool evaluation

Jon C. Muzio, PhD (Nottingham)
VLSI design and test, fault tolerant computing, design for testability, built-in self-test, multiple valued systems

Wendy Myrvold, PhD (Waterloo)
Graph theory, graph algorithms, network reliability, embedding graphs on surfaces, Latin squares, combinatorial algorithms

D. Dale Olesky, PhD (Toronto)
Linear algebra (especially matrix theory and combinational matrix analysis), numerical linear algebra, graph theory

Jianping Pan, PhD (Southeast, Nanjing)
Protocols for advanced networking, performance analysis of networked systems, applied network security

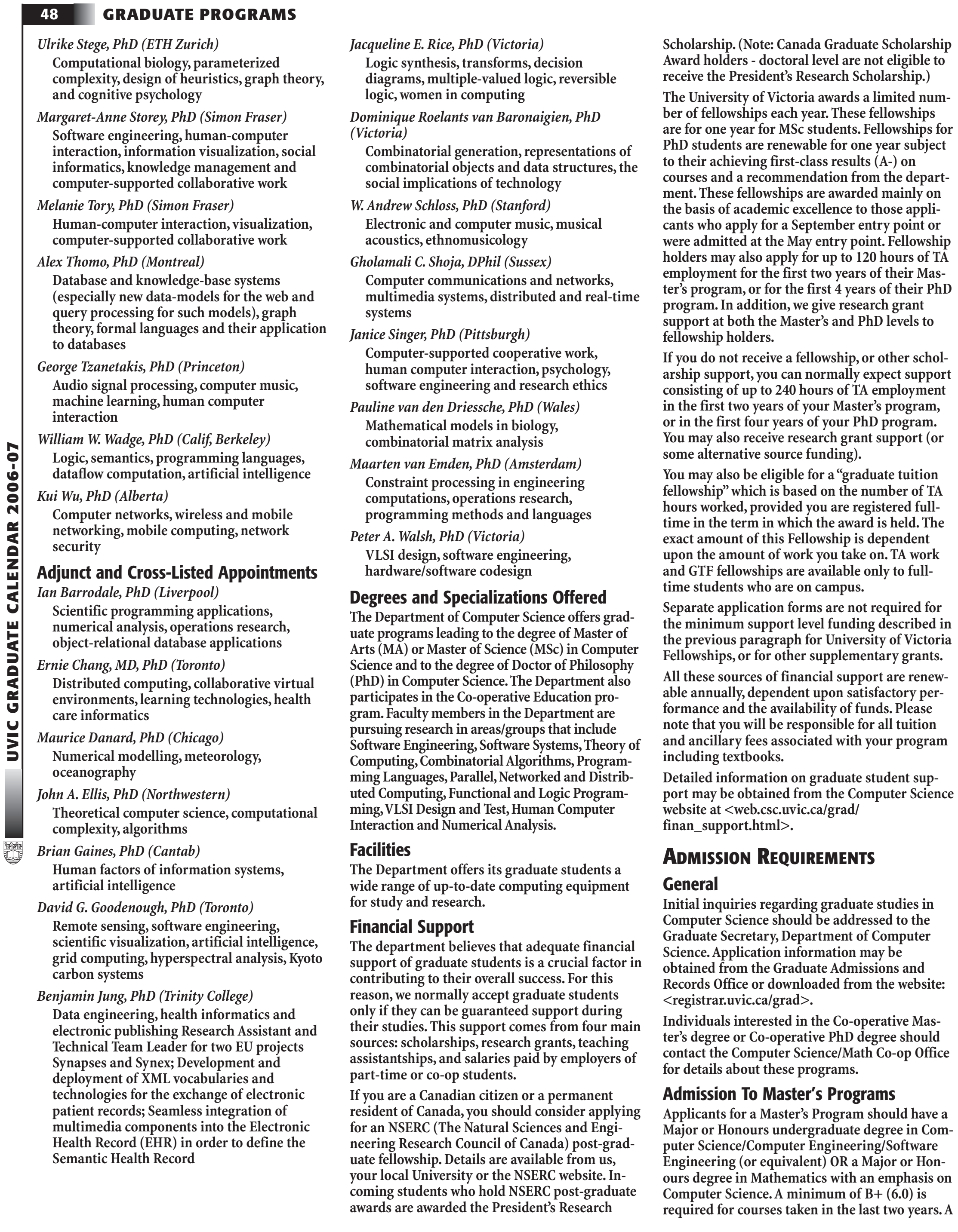
Frank D. K. Roberts, PhD (Liverpool)
Numerical analysis, approximation theory

Frank Ruskey, PhD (Calif, San Diego)
Combinatorial algorithms

Micaela Serra, PhD (Victoria)
Hardware/software co-design, VLSI design and test

Venkatesh Srinivasan, PhD (India)
Theory of computation, computational complexity theory

Graduate Programs, continued



Ulrike Stege, PhD (ETH Zurich)
Computational biology, parameterized complexity, design of heuristics, graph theory, and cognitive psychology

Margaret-Anne Storey, PhD (Simon Fraser)
Software engineering, human-computer interaction, information visualization, social informatics, knowledge management and computer-supported collaborative work

Melanie Tory, PhD (Simon Fraser)
Human-computer interaction, visualization, computer-supported collaborative work

Alex Thomo, PhD (Montreal)
Database and knowledge-base systems (especially new data-models for the web and query processing for such models), graph theory, formal languages and their application to databases

George Tzanetakis, PhD (Princeton)
Audio signal processing, computer music, machine learning, human computer interaction

William W. Wadge, PhD (Calif, Berkeley)
Logic, semantics, programming languages, dataflow computation, artificial intelligence

Kui Wu, PhD (Alberta)
Computer networks, wireless and mobile networking, mobile computing, network security

Adjunct and Cross-Listed Appointments

Ian Barrodale, PhD (Liverpool)
Scientific programming applications, numerical analysis, operations research, object-relational database applications

Ernie Chang, MD, PhD (Toronto)
Distributed computing, collaborative virtual environments, learning technologies, health care informatics

Maurice Danard, PhD (Chicago)
Numerical modelling, meteorology, oceanography

John A. Ellis, PhD (Northwestern)
Theoretical computer science, computational complexity, algorithms

Brian Gaines, PhD (Cantab)
Human factors of information systems, artificial intelligence

David G. Goodenough, PhD (Toronto)
Remote sensing, software engineering, scientific visualization, artificial intelligence, grid computing, hyperspectral analysis, Kyoto carbon systems

Benjamin Jung, PhD (Trinity College)
Data engineering, health informatics and electronic publishing Research Assistant and Technical Team Leader for two EU projects Synapses and Synex; Development and deployment of XML vocabularies and technologies for the exchange of electronic patient records; Seamless integration of multimedia components into the Electronic Health Record (EHR) in order to define the Semantic Health Record

Jacqueline E. Rice, PhD (Victoria)
Logic synthesis, transforms, decision diagrams, multiple-valued logic, reversible logic, women in computing

Dominique Roelants van Baronaigien, PhD (Victoria)
Combinatorial generation, representations of combinatorial objects and data structures, the social implications of technology

W. Andrew Schloss, PhD (Stanford)
Electronic and computer music, musical acoustics, ethnomusicology

Gholamali C. Shoja, DPhil (Sussex)
Computer communications and networks, multimedia systems, distributed and real-time systems

Janice Singer, PhD (Pittsburgh)
Computer-supported cooperative work, human computer interaction, psychology, software engineering and research ethics

Pauline van den Driessche, PhD (Wales)
Mathematical models in biology, combinatorial matrix analysis

Maarten van Emden, PhD (Amsterdam)
Constraint processing in engineering computations, operations research, programming methods and languages

Peter A. Walsh, PhD (Victoria)
VLSI design, software engineering, hardware/software codesign

Degrees and Specializations Offered

The Department of Computer Science offers graduate programs leading to the degree of Master of Arts (MA) or Master of Science (MSc) in Computer Science and to the degree of Doctor of Philosophy (PhD) in Computer Science. The Department also participates in the Co-operative Education program. Faculty members in the Department are pursuing research in areas/groups that include Software Engineering, Software Systems, Theory of Computing, Combinatorial Algorithms, Programming Languages, Parallel, Networked and Distributed Computing, Functional and Logic Programming, VLSI Design and Test, Human Computer Interaction and Numerical Analysis.

Facilities

The Department offers its graduate students a wide range of up-to-date computing equipment for study and research.

Financial Support

The department believes that adequate financial support of graduate students is a crucial factor in contributing to their overall success. For this reason, we normally accept graduate students only if they can be guaranteed support during their studies. This support comes from four main sources: scholarships, research grants, teaching assistantships, and salaries paid by employers of part-time or co-op students.

If you are a Canadian citizen or a permanent resident of Canada, you should consider applying for an NSERC (The Natural Sciences and Engineering Research Council of Canada) post-graduate fellowship. Details are available from us, your local University or the NSERC website. Incoming students who hold NSERC post-graduate awards are awarded the President's Research Scholarship. (Note: Canada Graduate Scholarship Award holders - doctoral level are not eligible to receive the President's Research Scholarship.)

The University of Victoria awards a limited number of fellowships each year. These fellowships are for one year for MSc students. Fellowships for PhD students are renewable for one year subject to their achieving first-class results (A-) on courses and a recommendation from the department. These fellowships are awarded mainly on the basis of academic excellence to those applicants who apply for a September entry point or were admitted at the May entry point. Fellowship holders may also apply for up to 120 hours of TA employment for the first two years of their Master's program, or for the first 4 years of their PhD program. In addition, we give research grant support at both the Master's and PhD levels to fellowship holders.

If you do not receive a fellowship, or other scholarship support, you can normally expect support consisting of up to 240 hours of TA employment in the first two years of your Master's program, or in the first four years of your PhD program. You may also receive research grant support (or some alternative source funding).

You may also be eligible for a "graduate tuition fellowship" which is based on the number of TA hours worked, provided you are registered full-time in the term in which the award is held. The exact amount of this Fellowship is dependent upon the amount of work you take on. TA work and GTF fellowships are available only to full-time students who are on campus.

Separate application forms are not required for the minimum support level funding described in the previous paragraph for University of Victoria Fellowships, or for other supplementary grants.

All these sources of financial support are renewable annually, dependent upon satisfactory performance and the availability of funds. Please note that you will be responsible for all tuition and ancillary fees associated with your program including textbooks.

Detailed information on graduate student support may be obtained from the Computer Science website at <web.csc.uvic.ca/grad/finan_support.html>.

ADMISSION REQUIREMENTS

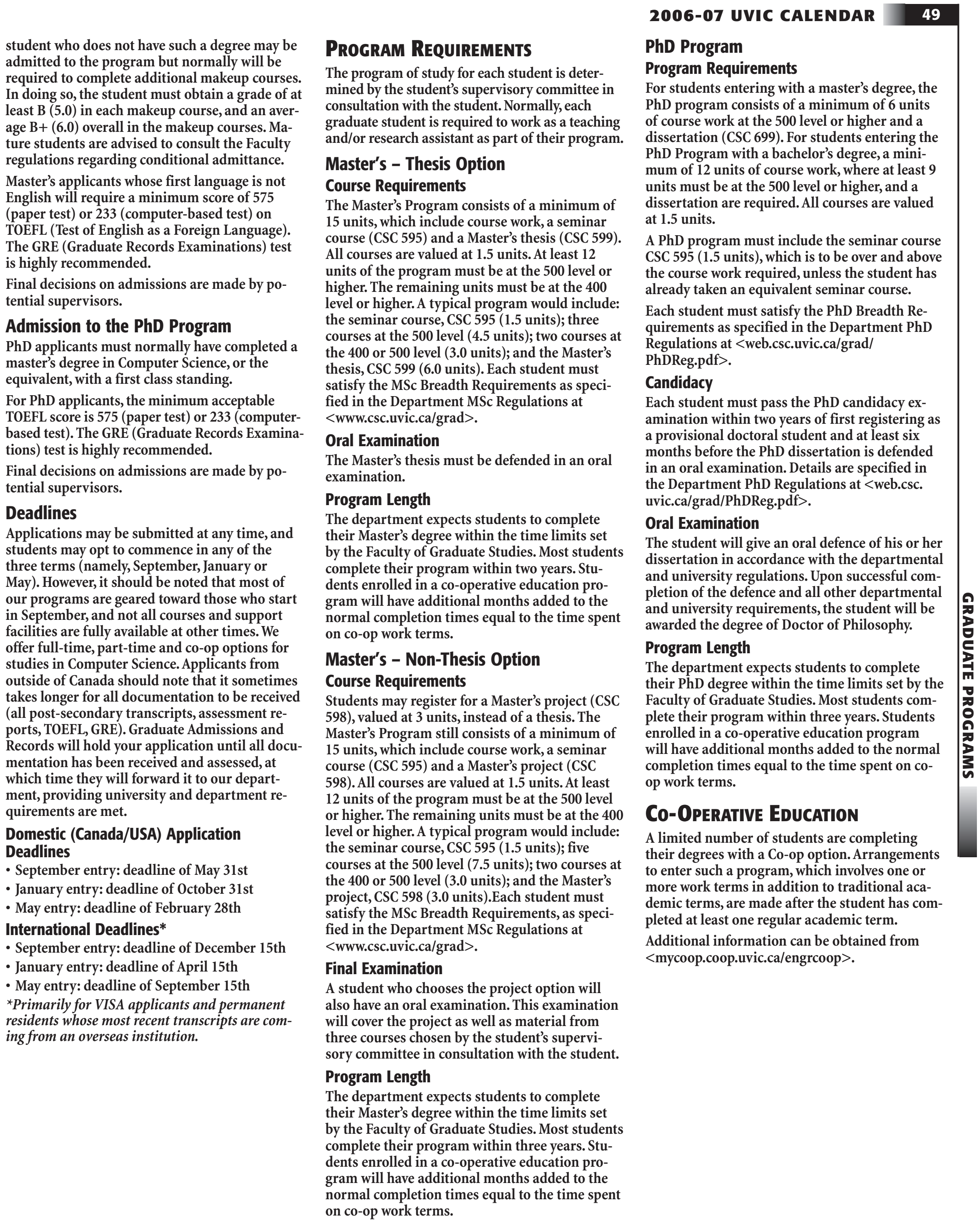
General

Initial inquiries regarding graduate studies in Computer Science should be addressed to the Graduate Secretary, Department of Computer Science. Application information may be obtained from the Graduate Admissions and Records Office or downloaded from the website: <registrar.uvic.ca/grad>.

Individuals interested in the Co-operative Master's degree or Co-operative PhD degree should contact the Computer Science/Math Co-op Office for details about these programs.

Admission To Master's Programs

Applicants for a Master's Program should have a Major or Honours undergraduate degree in Computer Science/Computer Engineering/Software Engineering (or equivalent) OR a Major or Honours degree in Mathematics with an emphasis on Computer Science. A minimum of B+ (6.0) is required for courses taken in the last two years. A



student who does not have such a degree may be admitted to the program but normally will be required to complete additional makeup courses. In doing so, the student must obtain a grade of at least B (5.0) in each makeup course, and an average B+ (6.0) overall in the makeup courses. Mature students are advised to consult the Faculty regulations regarding conditional admittance.

Master's applicants whose first language is not English will require a minimum score of 575 (paper test) or 233 (computer-based test) on TOEFL (Test of English as a Foreign Language). The GRE (Graduate Records Examinations) test is highly recommended.

Final decisions on admissions are made by potential supervisors.

Admission to the PhD Program

PhD applicants must normally have completed a master's degree in Computer Science, or the equivalent, with a first class standing.

For PhD applicants, the minimum acceptable TOEFL score is 575 (paper test) or 233 (computer-based test). The GRE (Graduate Records Examinations) test is highly recommended.

Final decisions on admissions are made by potential supervisors.

Deadlines

Applications may be submitted at any time, and students may opt to commence in any of the three terms (namely, September, January or May). However, it should be noted that most of our programs are geared toward those who start in September, and not all courses and support facilities are fully available at other times. We offer full-time, part-time and co-op options for studies in Computer Science. Applicants from outside of Canada should note that it sometimes takes longer for all documentation to be received (all post-secondary transcripts, assessment reports, TOEFL, GRE). Graduate Admissions and Records will hold your application until all documentation has been received and assessed, at which time they will forward it to our department, providing university and department requirements are met.

Domestic (Canada/USA) Application Deadlines

- September entry: deadline of May 31st
- January entry: deadline of October 31st
- May entry: deadline of February 28th

International Deadlines*

- September entry: deadline of December 15th
- January entry: deadline of April 15th
- May entry: deadline of September 15th

**Primarily for VISA applicants and permanent residents whose most recent transcripts are coming from an overseas institution.*

Program Requirements

The program of study for each student is determined by the student's supervisory committee in consultation with the student. Normally, each graduate student is required to work as a teaching and/or research assistant as part of their program.

Master's – Thesis Option

Course Requirements

The Master's Program consists of a minimum of 15 units, which include course work, a seminar course (CSC 595) and a Master's thesis (CSC 599). All courses are valued at 1.5 units. At least 12 units of the program must be at the 500 level or higher. The remaining units must be at the 400 level or higher. A typical program would include: the seminar course, CSC 595 (1.5 units); three courses at the 500 level (4.5 units); two courses at the 400 or 500 level (3.0 units); and the Master's thesis, CSC 599 (6.0 units). Each student must satisfy the MSc Breadth Requirements as specified in the Department MSc Regulations at <www.csc.uvic.ca/grad>.

Oral Examination

The Master's thesis must be defended in an oral examination.

Program Length

The department expects students to complete their Master's degree within the time limits set by the Faculty of Graduate Studies. Most students complete their program within two years. Students enrolled in a co-operative education program will have additional months added to the normal completion times equal to the time spent on co-op work terms.

Master's – Non-Thesis Option

Course Requirements

Students may register for a Master's project (CSC 598), valued at 3 units, instead of a thesis. The Master's Program still consists of a minimum of 15 units, which include course work, a seminar course (CSC 595) and a Master's project (CSC 598). All courses are valued at 1.5 units. At least 12 units of the program must be at the 500 level or higher. The remaining units must be at the 400 level or higher. A typical program would include: the seminar course, CSC 595 (1.5 units); five courses at the 500 level (7.5 units); two courses at the 400 or 500 level (3.0 units); and the Master's project, CSC 598 (3.0 units).Each student must satisfy the MSc Breadth Requirements, as specified in the Department MSc Regulations at <www.csc.uvic.ca/grad>.

Final Examination

A student who chooses the project option will also have an oral examination. This examination will cover the project as well as material from three courses chosen by the student's supervisory committee in consultation with the student.

Program Length

The department expects students to complete their Master's degree within the time limits set by the Faculty of Graduate Studies. Most students complete their program within three years. Students enrolled in a co-operative education program will have additional months added to the normal completion times equal to the time spent on co-op work terms.

PhD Program

Program Requirements

For students entering with a master's degree, the PhD program consists of a minimum of 6 units of course work at the 500 level or higher and a dissertation (CSC 699). For students entering the PhD Program with a bachelor's degree, a minimum of 12 units of course work, where at least 9 units must be at the 500 level or higher, and a dissertation are required. All courses are valued at 1.5 units.

A PhD program must include the seminar course CSC 595 (1.5 units), which is to be over and above the course work required, unless the student has already taken an equivalent seminar course.

Each student must satisfy the PhD Breadth Requirements as specified in the Department PhD Regulations at <web.csc.uvic.ca/grad/PhDReg.pdf>.

Candidacy

Each student must pass the PhD candidacy examination within two years of first registering as a provisional doctoral student and at least six months before the PhD dissertation is defended in an oral examination. Details are specified in the Department PhD Regulations at <web.csc.uvic.ca/grad/PhDReg.pdf>.

Oral Examination

The student will give an oral defence of his or her dissertation in accordance with the departmental and university regulations. Upon successful completion of the defence and all other departmental and university requirements, the student will be awarded the degree of Doctor of Philosophy.

Program Length

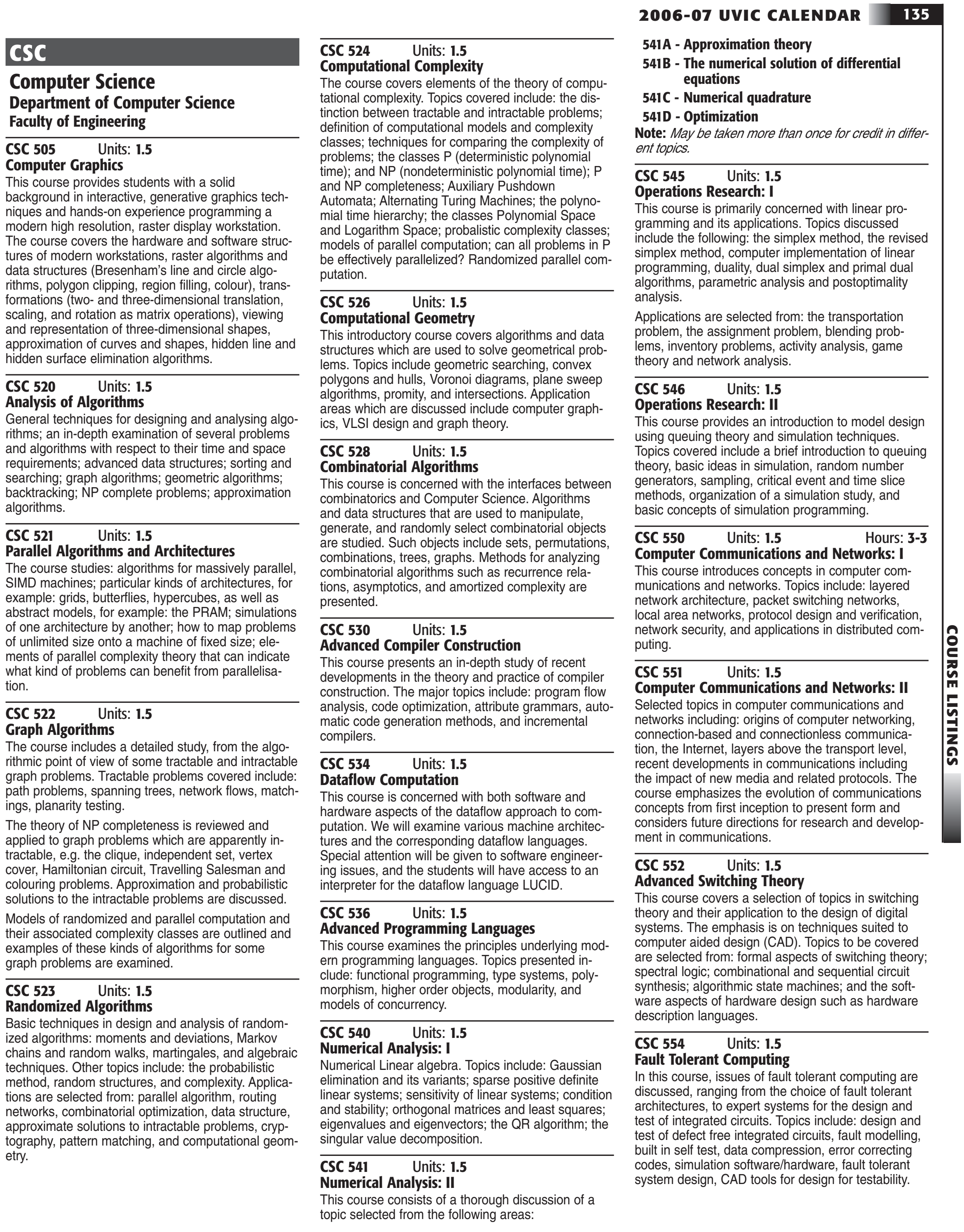
The department expects students to complete their PhD degree within the time limits set by the Faculty of Graduate Studies. Most students complete their program within three years. Students enrolled in a co-operative education program will have additional months added to the normal completion times equal to the time spent on co-op work terms.

Co-Operative Education

A limited number of students are completing their degrees with a Co-op option. Arrangements to enter such a program, which involves one or more work terms in addition to traditional academic terms, are made after the student has completed at least one regular academic term.

Additional information can be obtained from <mycoop.coop.uvic.ca/engrcoop>.

J.5 Graduate Course Descriptions



CSC

Computer Science

Department of Computer Science

Faculty of Engineering

CSC 505 Units: **1.5**
Computer Graphics
This course provides students with a solid background in interactive, generative graphics techniques and hands-on experience programming a modern high resolution, raster display workstation. The course covers the hardware and software structures of modern workstations, raster algorithms and data structures (Bresenham's line and circle algorithms, polygon clipping, region filling, colour), transformations (two- and three-dimensional translation, scaling, and rotation as matrix operations), viewing and representation of three-dimensional shapes, approximation of curves and shapes, hidden line and hidden surface elimination algorithms.

CSC 520 Units: **1.5**
Analysis of Algorithms
General techniques for designing and analysing algorithms; an in-depth examination of several problems and algorithms with respect to their time and space requirements; advanced data structures; sorting and searching; graph algorithms; geometric algorithms; backtracking; NP complete problems; approximation algorithms.

CSC 521 Units: **1.5**
Parallel Algorithms and Architectures
The course studies: algorithms for massively parallel, SIMD machines; particular kinds of architectures, for example: grids, butterflies, hypercubes, as well as abstract models, for example: the PRAM; simulations of one architecture by another; how to map problems of unlimited size onto a machine of fixed size; elements of parallel complexity theory that can indicate what kind of problems can benefit from parallelisation.

CSC 522 Units: **1.5**
Graph Algorithms
The course includes a detailed study, from the algorithmic point of view of some tractable and intractable graph problems. Tractable problems covered include: path problems, spanning trees, network flows, matchings, planarity testing.

The theory of NP completeness is reviewed and applied to graph problems which are apparently intractable, e.g. the clique, independent set, vertex cover, Hamiltonian circuit, Travelling Salesman and colouring problems. Approximation and probabilistic solutions to the intractable problems are discussed.

Models of randomized and parallel computation and their associated complexity classes are outlined and examples of these kinds of algorithms for some graph problems are examined.

CSC 523 Units: **1.5**
Randomized Algorithms
Basic techniques in design and analysis of randomized algorithms: moments and deviations, Markov chains and random walks, martingales, and algebraic techniques. Other topics include: the probabilistic method, random structures, and complexity. Applications are selected from: parallel algorithm, routing networks, combinatorial optimization, data structure, approximate solutions to intractable problems, cryptography, pattern matching, and computational geometry.

CSC 524 Units: **1.5**
Computational Complexity
The course covers elements of the theory of computational complexity. Topics covered include: the distinction between tractable and intractable problems; definition of computational models and complexity classes; techniques for comparing the complexity of problems; the classes P (deterministic polynomial time); and NP (nondeterministic polynomial time); P and NP completeness; Auxiliary Pushdown Automata; Alternating Turing Machines; the polynomial time hierarchy; the classes Polynomial Space and Logarithm Space; probalistic complexity classes; models of parallel computation; can all problems in P be effectively parallelized? Randomized parallel computation.

CSC 526 Units: **1.5**
Computational Geometry
This introductory course covers algorithms and data structures which are used to solve geometrical problems. Topics include geometric searching, convex polygons and hulls, Voronoi diagrams, plane sweep algorithms, promity, and intersections. Application areas which are discussed include computer graphics, VLSI design and graph theory.

CSC 528 Units: **1.5**
Combinatorial Algorithms
This course is concerned with the interfaces between combinatorics and Computer Science. Algorithms and data structures that are used to manipulate, generate, and randomly select combinatorial objects are studied. Such objects include sets, permutations, combinations, trees, graphs. Methods for analyzing combinatorial algorithms such as recurrence relations, asymptotics, and amortized complexity are presented.

CSC 530 Units: **1.5**
Advanced Compiler Construction
This course presents an in-depth study of recent developments in the theory and practice of compiler construction. The major topics include: program flow analysis, code optimization, attribute grammars, automatic code generation methods, and incremental compilers.

CSC 534 Units: **1.5**
Dataflow Computation
This course is concerned with both software and hardware aspects of the dataflow approach to computation. We will examine various machine architectures and the corresponding dataflow languages. Special attention will be given to software engineering issues, and the students will have access to an interpreter for the dataflow language LUCID.

CSC 536 Units: **1.5**
Advanced Programming Languages
This course examines the principles underlying modern programming languages. Topics presented include: functional programming, type systems, polymorphism, higher order objects, modularity, and models of concurrency.

CSC 540 Units: **1.5**
Numerical Analysis: I
Numerical Linear algebra. Topics include: Gaussian elimination and its variants; sparse positive definite linear systems; sensitivity of linear systems; condition and stability; orthogonal matrices and least squares; eigenvalues and eigenvectors; the QR algorithm; the singular value decomposition.

CSC 541 Units: **1.5**
Numerical Analysis: II
This course consists of a thorough discussion of a topic selected from the following areas:

- **541A - Approximation theory**
- **541B - The numerical solution of differential equations**
- **541C - Numerical quadrature**
- **541D - Optimization**

Note: *May be taken more than once for credit in different topics.*

CSC 545 Units: **1.5**
Operations Research: I
This course is primarily concerned with linear programming and its applications. Topics discussed include the following: the simplex method, the revised simplex method, computer implementation of linear programming, duality, dual simplex and primal dual algorithms, parametric analysis and postoptimality analysis.

Applications are selected from: the transportation problem, the assignment problem, blending problems, inventory problems, activity analysis, game theory and network analysis.

CSC 546 Units: **1.5**
Operations Research: II
This course provides an introduction to model design using queuing theory and simulation techniques. Topics covered include a brief introduction to queuing theory, basic ideas in simulation, random number generators, sampling, critical event and time slice methods, organization of a simulation study, and basic concepts of simulation programming.

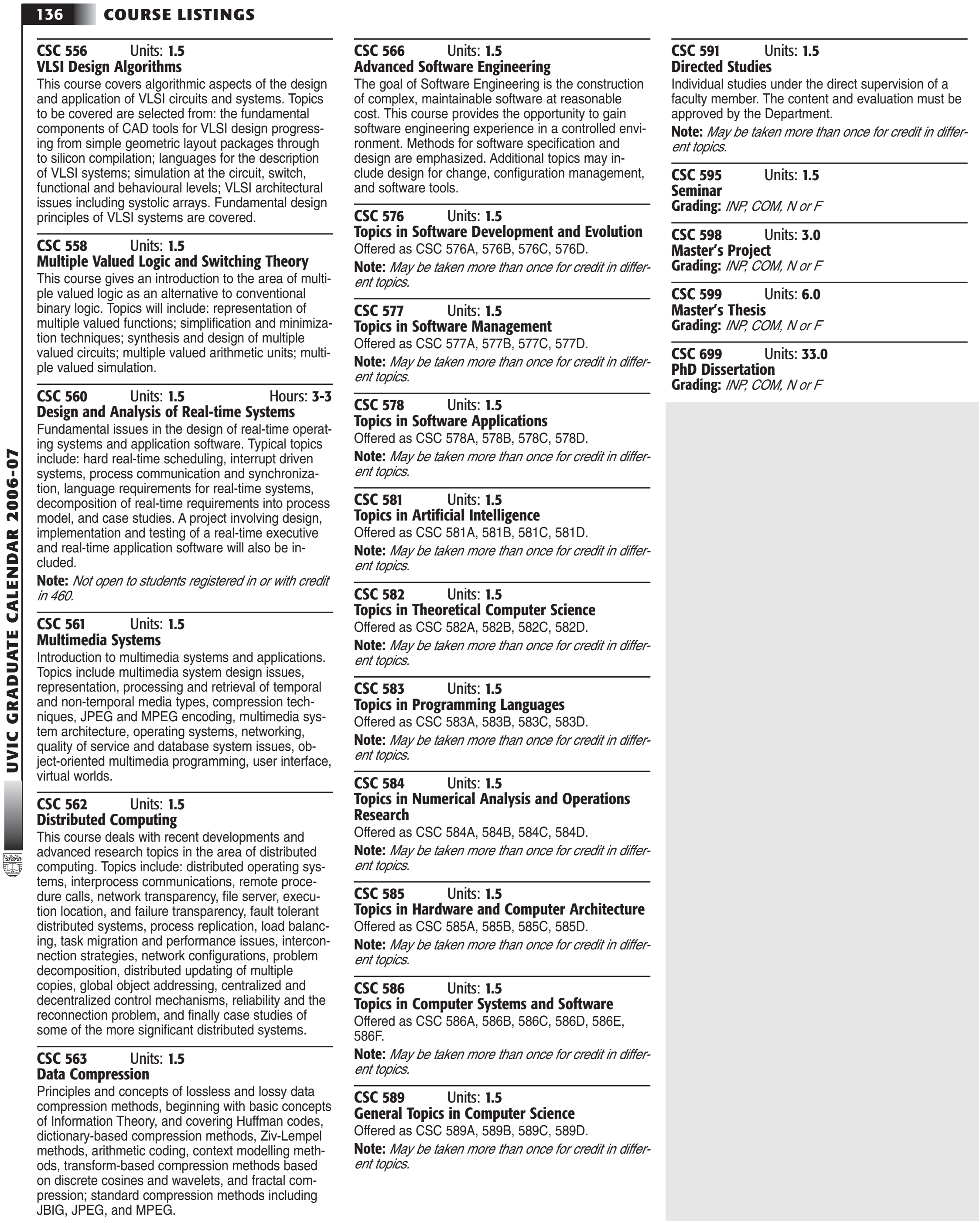
CSC 550 Units: **1.5** Hours: **3-3**
Computer Communications and Networks: I
This course introduces concepts in computer communications and networks. Topics include: layered network architecture, packet switching networks, local area networks, protocol design and verification, network security, and applications in distributed computing.

CSC 551 Units: **1.5**
Computer Communications and Networks: II
Selected topics in computer communications and networks including: origins of computer networking, connection-based and connectionless communication, the Internet, layers above the transport level, recent developments in communications including the impact of new media and related protocols. The course emphasizes the evolution of communications concepts from first inception to present form and considers future directions for research and development in communications.

CSC 552 Units: **1.5**
Advanced Switching Theory
This course covers a selection of topics in switching theory and their application to the design of digital systems. The emphasis is on techniques suited to computer aided design (CAD). Topics to be covered are selected from: formal aspects of switching theory; spectral logic; combinational and sequential circuit synthesis; algorithmic state machines; and the software aspects of hardware design such as hardware description languages.

CSC 554 Units: **1.5**
Fault Tolerant Computing
In this course, issues of fault tolerant computing are discussed, ranging from the choice of fault tolerant architectures, to expert systems for the design and test of integrated circuits. Topics include: design and test of defect free integrated circuits, fault modelling, built in self test, data compression, error correcting codes, simulation software/hardware, fault tolerant system design, CAD tools for design for testability.

Graduate Course Descriptions, continued



136 **COURSE LISTINGS**

UVIC GRADUATE CALENDAR 2006-07

CSC 556 Units: **1.5**
VLSI Design Algorithms
This course covers algorithmic aspects of the design and application of VLSI circuits and systems. Topics to be covered are selected from: the fundamental components of CAD tools for VLSI design progressing from simple geometric layout packages through to silicon compilation; languages for the description of VLSI systems; simulation at the circuit, switch, functional and behavioural levels; VLSI architectural issues including systolic arrays. Fundamental design principles of VLSI systems are covered.

CSC 558 Units: **1.5**
Multiple Valued Logic and Switching Theory
This course gives an introduction to the area of multiple valued logic as an alternative to conventional binary logic. Topics will include: representation of multiple valued functions; simplification and minimization techniques; synthesis and design of multiple valued circuits; multiple valued arithmetic units; multiple valued simulation.

CSC 560 Units: **1.5** Hours: **3-3**
Design and Analysis of Real-time Systems
Fundamental issues in the design of real-time operating systems and application software. Typical topics include: hard real-time scheduling, interrupt driven systems, process communication and synchronization, language requirements for real-time systems, decomposition of real-time requirements into process model, and case studies. A project involving design, implementation and testing of a real-time executive and real-time application software will also be included.
Note: *Not open to students registered in or with credit in 460.*

CSC 561 Units: **1.5**
Multimedia Systems
Introduction to multimedia systems and applications. Topics include multimedia system design issues, representation, processing and retrieval of temporal and non-temporal media types, compression techniques, JPEG and MPEG encoding, multimedia system architecture, operating systems, networking, quality of service and database system issues, object-oriented multimedia programming, user interface, virtual worlds.

CSC 562 Units: **1.5**
Distributed Computing
This course deals with recent developments and advanced research topics in the area of distributed computing. Topics include: distributed operating systems, interprocess communications, remote procedure calls, network transparency, file server, execution location, and failure transparency, fault tolerant distributed systems, process replication, load balancing, task migration and performance issues, interconnection strategies, network configurations, problem decomposition, distributed updating of multiple copies, global object addressing, centralized and decentralized control mechanisms, reliability and the reconnection problem, and finally case studies of some of the more significant distributed systems.

CSC 563 Units: **1.5**
Data Compression
Principles and concepts of lossless and lossy data compression methods, beginning with basic concepts of Information Theory, and covering Huffman codes, dictionary-based compression methods, Ziv-Lempel methods, arithmetic coding, context modelling methods, transform-based compression methods based on discrete cosines and wavelets, and fractal compression; standard compression methods including JBIG, JPEG, and MPEG.

CSC 566 Units: **1.5**
Advanced Software Engineering
The goal of Software Engineering is the construction of complex, maintainable software at reasonable cost. This course provides the opportunity to gain software engineering experience in a controlled environment. Methods for software specification and design are emphasized. Additional topics may include design for change, configuration management, and software tools.

CSC 576 Units: **1.5**
Topics in Software Development and Evolution
Offered as CSC 576A, 576B, 576C, 576D.
Note: *May be taken more than once for credit in different topics.*

CSC 577 Units: **1.5**
Topics in Software Management
Offered as CSC 577A, 577B, 577C, 577D.
Note: *May be taken more than once for credit in different topics.*

CSC 578 Units: **1.5**
Topics in Software Applications
Offered as CSC 578A, 578B, 578C, 578D.
Note: *May be taken more than once for credit in different topics.*

CSC 581 Units: **1.5**
Topics in Artificial Intelligence
Offered as CSC 581A, 581B, 581C, 581D.
Note: *May be taken more than once for credit in different topics.*

CSC 582 Units: **1.5**
Topics in Theoretical Computer Science
Offered as CSC 582A, 582B, 582C, 582D.
Note: *May be taken more than once for credit in different topics.*

CSC 583 Units: **1.5**
Topics in Programming Languages
Offered as CSC 583A, 583B, 583C, 583D.
Note: *May be taken more than once for credit in different topics.*

CSC 584 Units: **1.5**
Topics in Numerical Analysis and Operations Research
Offered as CSC 584A, 584B, 584C, 584D.
Note: *May be taken more than once for credit in different topics.*

CSC 585 Units: **1.5**
Topics in Hardware and Computer Architecture
Offered as CSC 585A, 585B, 585C, 585D.
Note: *May be taken more than once for credit in different topics.*

CSC 586 Units: **1.5**
Topics in Computer Systems and Software
Offered as CSC 586A, 586B, 586C, 586D, 586E, 586F.
Note: *May be taken more than once for credit in different topics.*

CSC 589 Units: **1.5**
General Topics in Computer Science
Offered as CSC 589A, 589B, 589C, 589D.
Note: *May be taken more than once for credit in different topics.*

CSC 591 Units: **1.5**
Directed Studies
Individual studies under the direct supervision of a faculty member. The content and evaluation must be approved by the Department.
Note: *May be taken more than once for credit in different topics.*

CSC 595 Units: **1.5**
Seminar
Grading: *INP, COM, N or F*

CSC 598 Units: **3.0**
Master's Project
Grading: *INP, COM, N or F*

CSC 599 Units: **6.0**
Master's Thesis
Grading: *INP, COM, N or F*

CSC 699 Units: **33.0**
PhD Dissertation
Grading: *INP, COM, N or F*