0. INTRODUCTION

Our goal in the Department of Computer Science is to be outstanding in every aspect of our function, namely our research, our undergraduate and graduate programs and our outreach activities to the rest of the community. We strive to be an exciting place to learn and to work, where no goal is too ambitious to strive for, where nothing is too sacred to laugh at, and where everybody’s ideas count.

The department plays a leadership role in the Canadian and international computer science communities. Leadership areas include research, curriculum, technology development, policy and professional practice. We also are prepared to play a leadership role in helping other areas of academia and society to integrate advances in information technology into their endeavours.

The CS Department at UBC has long had the reputation as being of high quality in both teaching and research, and of being very collegial and warm spirited. These facts were very much in evidence in our visit. This department certainly ranks within the top three in Canada and within the top 25 in North America from a research perspective. This is an impressive achievement considering that this is a mid-sized department that is smaller in complement than, for example, the Universities of Waterloo and Toronto.


0.1 Brief History

The Department of Computer Science was created at UBC in 1968. It grew from an initial six faculty members to fifteen full-time faculty in the mid 1980's. In 1988, as the result of a BC provincial government initiative targeting research in information technology and related fields at the (then) three provincial universities, and a concurrent commitment from the senior administration at UBC to expand the computer science department into a world-ranked department, an ambitious program of growth and strengthening of computer science at UBC was launched. Five years later the faculty complement had reached twenty-nine, an expansion which, when measured by the excellence of the new recruits as well as their number, ranked among the most significant in the world.

The department's faculty complement remained essentially flat for the next seven years. During the period 1993-2000, eight new faculty members joined the department and seven left. All but one of the seven who left were tenured faculty. Two of the eight new faculty appointments were at a senior level. The other six were junior. Over the two year period 1998/2000, Barry McBride, Vice-President Academic, created seven new faculty positions split equally between Computer Science and Electrical and Computer Engineering, with at least one appointment to be joint.
The department continues to recruit aggressively. A further four new junior faculty members were appointed in 2000 (one joint with Psychology) and three others joined us early in 2001. We note, with deep regret, the death in 2000 of Alain Fournier, a senior faculty member in computer graphics.

1. RETENTION AND RENEWAL OF FACULTY AND STAFF

The Department has lost many key young faculty and is on the brink of losing more. Specifically, it lost four people last year, and seven in the past five. Many of these people are already internationally renowned researchers, making the loss all the more serious. If action is not taken within three months, it will likely lose several more people. (emphasis added) This is particularly galling since the Department was the envy of the computer science world in the late eighties and early nineties for its remarkably good hires.

- Evaluation of the Department of Computer Science at the University of British Columbia, Interim Report.

1.1 Hiring/Recruitment Plans.
Since 1996 we have been working within the framework of an approved Department Hiring Plan. Circumstances, however, have changed. Significant new resources were committed to the department in June, 2000 in response to a crisis in faculty retention. Outstanding hiring is indeed taking place. However, as noted, increases have been offset by significant losses (seven due to resignations and one due to death). Losses included mid career and senior researchers in areas of traditional departmental strength. Of necessity, our Department Hiring Plan remains a living document able to respond to unforeseen circumstances and able to pursue unforeseen opportunities.

As of January, 2001, the department has a full-time faculty complement of 34.5, including 29.5 research faculty and 5 instructors. One instructor (Murray Goldberg) is on extended leave of absence. Murray is the founder of WebCT and President of WebCT/Canada. Based on commitments and plans already in place, our most important recommendations for the next five years can be summarized as follows:

- increase the tenured/tenure-track faculty complement from 34.5 to 46 by
  - increasing the number of research faculty from 29.5 to 38
  - increasing the number of instructors from 5 to 8;
- increase base budget (i.e., GPOF) funding for technical and clerical/administrative staff by 4 and 5 positions respectively;
- maintain strength in core research areas while rectifying obvious imbalances among them;
- aggressively recruit in our areas of most critical need: systems (including networks and operating systems), databases, and graphics/human-computer interfaces;
- support faculty recruitment with funding for start up packages, nominally $60K-$90K per new faculty member, with experimental areas identified as critical need (eg. systems) requiring significantly larger amounts ($150K-$250K or more); and
- maintain flexibility in our hiring strategy to remain competitive and opportunistic.
Further, in collaboration with other units, the department will pursue other initiatives that enhance the department’s leadership position in research, curriculum and technology development in information technology, broadly defined.

1.2 Department Strengths
Our department has considerable strength in several research areas:
- Computational Intelligence (Conati, Hoos, Little, Lowe, Mackworth, MacLean, Pai, Poole, Rensink, Rosenberg, Woodham)
- Databases (Ng, Tsiknis)
- Educational Technologies (Cavers, Conati, Goldberg, Klawe, Tsiknis)
- Graphics and Human-computer Interfaces (Booth, Heidrich, MacLean, Rensink)
- Integrated Hardware/Software Systems (Greenstreet, Hu)
- Operating Systems and Networks (Feeley, Hutchinson, Vuong, Wagner)
- Scientific Computation and Visualization (Ascher, Carter, Varah)
- Social, Legal and Economic Impacts of Computers (Rosenberg)
- Software Engineering and Programming Languages (De Volder, Kiczales, Murphy)
- Theoretical Aspects of Computer Science (Belleville, Condon, Evans, Hoos, Kirkpatrick, Klawe, Pippenger, Wagner)

In what follows the proposed areas of hiring are annotated with the names of Faculty of Science Research Themes. These themes, the original 12 Science CRC themes as submitted in March 2000, reflect hiring proposals from Science aggregated into themes:

- Bioinformatics and Functional Genomics
- Biophysics of Macromolecular Assemblies
- Cognitive Systems
- Computation Modeling Analysis
- Environment
- Fundamental Mathematics
- Health
- Hybrid Systems
- Information Technology
- Origins of Matter and The Universe
- Quantum Computing
- Risk and Mathematical Finance

1.3 Areas of Critical Need
It should be noted that certain areas are designated as core areas for Computer Science in addition to their labeling as themes, as described above. The following areas are critical growth areas for the department:
- *databases* Core and Theme: Information Technology
- *graphics and human-computer interfaces* Core and Theme: Information Technology and Cognitive Systems
- *systems*: (including networks, operating systems, software engineering, and computer architecture) Core and Theme: Information Technology

1.4 Areas Targeted for Short-term Growth
Areas that have been identified as important for short-term growth include:

- **Agents**
  Agent-based computing is becoming prevalent in many e-commerce and e-business applications, and depends on and enriches diverse areas such as systems, computational intelligence and robotics.
  Core and Theme: Information Technology and Cognitive Systems

- **Bioinformatics**
  Computational methods are becoming increasingly important in the context of research and development in molecular biology and biochemistry. Bioinformatics is a fast-growing and highly interdisciplinary field that is focused on computational problems in molecular biology and their algorithmic solution, as well as with biomolecular computation, i.e., the use of biomolecules and biomolecular techniques for performing computations in a massively parallel way.
  Theme: Bioinformatics and Functional Genomics

- **Cognitive Systems**
  A new discipline for studying the mind/brain in humans, animals and artificial systems. LCI, Imager and E-GEMS all have outstanding records in this area that complement programs in Cognitive Systems.
  Core and Theme: Information Technology and Cognitive Systems

- **Computation, Music and Sound**
  There is an increasing and fundamental interest in computational techniques for creating, analysing, and manipulating sound and music. Driven by application needs as well as deep fundamental questions, this inherently interdisciplinary field comprises topics such as acoustic virtual environments, auditory user interfaces, computer-assisted or automated composition, electronic music publishing, music and sound information retrieval, music representation, and synthesis.
  Theme: Information Technology, Cognitive Systems

- **Computer Architecture**
  Computer architecture is central to computer science and engineering. In computer architecture, the models of computation arising from theoretical computer science and programming languages meet the physical laws that determine feasible implementations.
  Computer architecture provides the infrastructure required for research in all applied areas of computer science including operating systems, embedded systems, and tangible computing.
  Core and Theme: Information Technology

- **Educational Technology**
  Building on existing research excellence (E-GEMS, WebCT, Roadmap to Computing, LCI, Imager), CS has the opportunity to build an excellent research group that will devise innovative ways in which computers can enhance education, learning and training.
  Theme: Cognitive Systems
• **Empirical Algorithmics**
In many areas of computer science, experiments are the primary means of demonstrating the value and potential of new techniques. It is increasingly widely being recognized that appropriate empirical approaches often yield new insights into the behaviour of algorithms and systems. Advanced empirical methods for analysing and comparing algorithms or systems have been crucial in the development of the best known algorithms for many hard problems, in particular in fields like Artificial Intelligence or Bioinformatics.
Theme: Information Technology, Computation Modeling Analysis

• **Geographic Information Systems**
Geographic information systems (GIS) support research in a broad range of earth and ocean sciences as well as decision making in government and industry. Research into GIS involves a multitude of disciplines, especially Geography, Forestry and Computer Science, where computational geometry, databases, remote sensing and image understanding, among others, play essential roles.
Core and Theme: Information Technology and Environment

• **Hybrid Systems**
With advances in microelectronics, digital microprocessors are being employed in an increasingly wide variety of applications, often replacing analog electronics or mechanical feedback mechanisms. The resulting hybrid system has a digital, software program supervising continuous, physical process. Research in hybrid systems is multidisciplinary including Computer Science, Control Engineering, and Mathematics, especially in the areas of hybrid control, hybrid verification, numerical optimization and software engineering.
Core and Theme: Hybrid Systems

• **Intelligent Decision-Making for Multi-Agent Systems**
Automated decision making for multi-agent systems considers what an agent should do based on its capabilities, its preferences, its observations of the environment, its background knowledge, its past experiences and what the other agents are doing. Automated decision making has many potential applications, for example in electronic commerce, medical decision-making, public policy and adaptive interfaces. UBC has the potential to build on isolated strengths in the various disciplines such as Computer Science, Commerce and Statistics.
Theme: Information Technology

• **Numerical Methods and Scientific Computation**
A strength in Computer Science that supports a variety of other areas including hybrid systems, verification, simulation and robotics. New hiring in this area is needed to support growth and to anticipate future retirements in this area.
Core and Theme: Information Technology and Computation Modeling Analysis

• **Quantum Computation**
During the past two decades, there has been growing awareness that computers operating on quantum mechanical principles might be able to carry out, within reasonable time, computations that would be completely infeasible with conventional "classical" computers.
Theme: Quantum Computation

- Security and Cryptography
  Security is an aspect of networks and operating systems that is becoming increasingly important with highly interconnected systems. Cryptography has an important role as a security tool; it arises out of more theoretical investigations and merges with the protocol and architectural methodology of security.
  Theme: Information Technology and Quantum Computation

- Social and Legal Aspects of Computing
  This area concerns itself with the identification and exploration of the host of issues associated with the new information technologies: privacy, free speech (content regulation), access to the technologies, intellectual property rights, and the transformation of government services and information resources.
  Core and Theme: Information Technology

- Software Engineering
  A field of computer science and engineering that focuses on the problems and challenges of building a multi-version software system with multiple people. One critical area in which we need to hire involves the verification and validation of software, to ensure the right system is built in the right way.
  Core and Theme: Information Technology

- Tangible Computing
  Emerging as a new core of ideas and techniques essential for natural human interface devices, interactive workspaces, personal robots, interactive toys, and physical simulations in virtual environments, games, and animation. UBC has exceptional potential to lead in this area, with strengths in constituent areas in Computer Science (Computational Intelligence, Imager, MAGIC, and Scientific Computation and Visualization) and Electrical and Computer Engineering, and new hires will play a large role in the New Media Innovation Centre and the proposed Institute for Computing, Information and Cognitive Systems.
  Core and Theme: Information Technology and Cognitive Systems

2. THE LEARNING ENVIRONMENT

This Department has very little control over enrollments in its own courses. Furthermore, enrollments and departmental revenue are essentially independent. . .We do not believe enrolment reductions are the answer, but we strongly recommend significantly increasing the resources to the Department consistent both with the enrolment increases it has seen in the past three years, and the even higher expected enrolments in future.

- Evaluation of the Department of Computer Science at the University of British Columbia, Interim Report.

2.0 Introduction
In this section, the Learning Environment will be discussed under three headings: the undergraduate component, the graduate component, and the multidisciplinary component. All of
these areas are growing and offer increasingly difficult challenges. For example, the undergraduate curriculum can be characterized by dividing its responsibilities into the following three areas:

- Core Computer Science
- Multidisciplinary contributions
- Service Computer Science

As will be seen below, the Department is engaged in a wide range of exciting activities that hold great promise for the University and society at large.

2.1 Structure of the Undergraduate Learning Environment

Computer Science has a rich theoretical foundation complemented by complex synthetic and experimental aspects whose scale demands novel design methodologies the formulation of which is a research area in its own right. Applications and interdisciplinary connections (such as cognitive systems and bioinformatics) increasingly require a deep understanding of other disciplines as well as an appreciation of their often profound social implications. The primary challenge of computer science education is to convey the richness of these various dimensions of our discipline, to expose students to current research, and to prepare them for the inevitable changes to come.

The Department is committed to maintaining an environment that fosters the highest possible levels of quality learning by encouraging active learner's participation, new delivery methods and better communication. We are committed to

- train undergraduate students, graduate students and post-doctoral fellows to meet the needs of industry and academia (research and teaching), with a sufficient versatility to permit adaptation to and leadership in the inevitable changes of the future,
- play a leading role in the increasing integration of computing and information technology into the teaching and research activities of virtually all sectors of the university (including, notably, the development of technologies for innovative teaching and learning, as well as major cross-disciplinary research initiatives.)

The rapid and dramatic changes associated with the continued growth and diversity of computer and information technology have significant implications for our undergraduate program. We identify the following challenges for the near future.

2.1a Class Sizes and Admission Control

The number of students in the Major and Honours programs in Computer Science (CS) has increased from 520 in the 1996 Winter session to 997 in the 2000 Winter session. Such demand is expected to continue and grow in the next five years. The demand comes both for CS majors (IT graduates) as envisioned in the draft ITBC proposals, and from the growing University-wide role of IT. Increasingly UBC students outside our department and faculty will be required by their (major, minor, or honours) programs to include CS courses. For example, the Life Sciences have recently suggested a CS course as part of the basic requirements, while more programs in the Faculty of Science are expected to have significant Computer Science components in the
future. In Applied Science, a number of programs require Computer Science courses and the Faculty has recently proposed an Information Technology minor program.

The most obvious (and perhaps most serious) consequence of our recent expansion in undergraduate enrolments is a dramatic increase in almost all undergraduate class sizes. Despite the strict enrolment requirements, a number of first, second, third, and fourth year classes in 1999-00 had 205, 185, 200, and 160 students, respectively.

We need to improve both the student-to-instructor and student-to-teaching-assistant ratios, by reducing class and lab sizes, especially in the third and fourth year. Our target is to reduce the maximum class sizes to 180, 150, 120, 80 students in the corresponding years, while keeping some third and fourth year classes at sizes much lower than the maximum sizes. This will provide students with the opportunity to experience closer interactions with the instructors and other classmates, and improve the learning process. This is especially important for our honours students, who currently do not have this experience.

To reach our target class sizes and to sustain the teaching quality we desire, in addition to acquiring new resources, the Department needs to have control over admission to the Computer Science programs. The current mechanisms of enrollment restrictions do not provide the appropriate control to program and course sizes. The current rules of the Faculty of Science allow any student in Science to select a specialization at the beginning of the second year. It is very difficult to plan our future offerings without an accurate estimate on the number of students who are planning to enroll in our programs. A comprehensive admission procedure is the only mechanism that will allow us to plan our course offerings ahead and optimize the available resources. It will also permit us to serve the general student community better by reserving seats in our courses for students from other departments and faculties.

2.1b Curriculum Development and Revision

The quick rate of changes in the field of Computer Science presents unique challenges with respect to curriculum development. While the core set of principles and basic concepts is well established, the methods of using these principles to build solutions are mutating rapidly. Frequent updates are needed in most courses, in order to present the concepts to the students in a context that is as current as possible. For instance, our operating system courses can not merely refer to the versions of the operating systems that existed five or ten years ago, but should also refer to the current extensions of these, and to newer operating systems that are only emerging, but will be commonplace by the time today’s students graduate.

In contrast, the material and methodology used in other lower-level science courses are much better established, and do not vary nearly as much. Consequently, we need to spend, proportionally, more time updating our curriculum than some other departments. Our enrollment problems are having a serious impact on these updates, as we are more occupied dealing with large classes and the corresponding administrative problems than with curriculum issues. Therefore, it is important to find ways and resources that permit frequent revisions to our programs and individual courses to ensure compliance with internationally recognized curriculum models in our field.

The revision of our first and second year curriculum is one of our first priorities. It is very important that our lower level courses be up to date as they provide all the foundations for the upper level courses. Independent updates of these courses by their instructors may not be sufficient to ensure smooth transition of our students into upper level courses. It is imperative
that the Department review and update the entire lower-level curriculum, at least once every five years. Such a revision is long overdue and needs to be carried out as soon as it becomes feasible.

2.1c Honours Courses
The Computer Science Department currently offers Honours and Combined Honours programs to qualified undergraduate students. These programs allow Honours students to experience research during their undergraduate studies by attending an Honours Research Seminar (CPSC349) during their 3rd year and completing an Honours Thesis (CPSC449) in their 4th year.

To better prepare our Honours students for graduate studies we need to design more honours courses which would include advanced material that is usually available in some graduate courses. Moreover, it is important that we keep the honours classes small enough to provide an intimate environment that fosters the necessary teacher-student interactions.

More specifically, to improve our Honours program we need to

- develop separate core courses for our Honours students,
- redevelop some of our 3rd and 4th year courses, currently offered to both Major and Honours students, to offer a more challenging "Honours version", whose material would delve more deeply into current research,
- develop special capstone courses for this program that draw on the knowledge acquired from a broad range of other upper-level courses,
- keep the sizes of the Honours classes and labs as small as possible.

To satisfy the needs outlined above for the Computer Science Honours program, in addition to faculty resources, we need further financial resources that will allow us to:

- purchase special lab equipment,
- expand current lab space, and
- employ highly qualified Teaching Assistants for the courses offered to our Honours students.

2.1d New Program Options
Until recently, the Department was offering four programs: a CPSC Major, a CPSC Honours, and two combined Honours programs with Mathematics and Physics. To respond to the increasing demand for graduates with thorough knowledge of software development methodologies, the Department has recently designed and offered a Software Engineering option for its Major and Honours programs. Electrical and Computer Engineering also offers a Software Engineering option. Both departments are understaffed to teach these options. Our software engineering curricula include 5 undergraduate courses and 3 graduate courses, but we have only three faculty in this area. We are confident in our ability to recruit additional faculty because of the high quality of our current professors and the support of our NSERC/Xerox Canada/Sierra Systems Industrial Research Chair in Software Design.

As the scope of Computer Science expands and new areas mature, the need for expanding our curriculum increases. Such expansion usually entails the addition of new courses that introduce new topics, and the design of new options to the core Computer Science programs. The demand for such options comes from both industry and the academic community. The
Department should have the resources to quickly respond to such needs in order to maintain its competitiveness and reputation.

Taking into account the current directions of our field, we should acquire sufficient resources to develop the following options for our Major and Honours program, in the next five years:

- Bioinformatics,
- Computational Statistics and Information Retrieval,
- Computer Systems and Networks,
- Electronic Commerce (e-commerce),
- Human Computer Interaction and New Media, and
- Scientific Computation

These options will increase our collaboration with other Departments and Faculties in multidisciplinary programs. We have recently participated in the design of an undergraduate Major in Cognitive Systems jointly with the Departments of Linguistics, Philosophy, and Psychology. There are collaboration opportunities with Microbiology, Commerce, Mathematics, Statistics and other departments that we wish to explore.

2.1e Service Courses
Several departments require their students to take one or more Computer Science courses, and this trend will increase in the next few years. For instance, the Department of Electrical and Computer Engineering has recently introduced a Software-Engineering Option that requires several Computer Science courses. These courses are already in very high demand, and the pressure on them will increase markedly.

Unfortunately, once established, we have very little control over enrollment in these other programs. The most acute example is courses that we teach specifically for students in the Faculty of Applied Science (CPSC 152, CPSC 252, and soon CPSC 352), or other courses that are required or recommended by Applied Science programs (CPSC 310 up to this year, as well as CPSC 304, 315, 318, and a few others).

We have been unsuccessful in obtaining timely information regarding the expected number of Applied Science students that will need to enroll into these courses. The information that is communicated often underestimates the actual enrollment. As a result, we are unable to plan effectively, and we are often forced to adopt unsatisfactory last-minute solutions that generate resentment from our students who are denied access to these courses and to other courses that use the same laboratory facilities.

Having accurate and timely information regarding enrollment in the service courses would allow us to plan ahead, and avoid these problems. It is thus imperative that we establish new enrollment agreements and effective communication mechanisms with other Departments and Faculties, which require our courses.

2.1f Distance Education
With the increasing popularity of IT courses and the limited seats in the traditional classrooms, the demand for Guided Independent Study (GIS) courses in Computer Science is rising. GIS courses are currently owned, administered and delivered by the Distance Education division of Continuing Studies. CPSC 315, our introductory course to Operating Systems is the only course
in Computer Science that has been offered in that format. The Department would be willing to
design and offer more GIS courses if the following concerns are addressed:

- An academic department should be responsible for the course, and must approve the
course instructor. This ensures that the standards used for our course instructors will
apply to GIS courses.
- The responsible department should regularly update the GIS material to match that of the
traditional offerings of the course. This task should be funded by the unit that offers the
course.
- Approval of GIS courses should follow the standard curriculum approval procedure
established for the traditional courses.
- Students should have the opportunity to complete formal course and teaching evaluations
for the instructor and the teaching assistants of a GIS course. Such evaluations are
essential to monitor and protect the quality of education being offered.
- Prerequisite requirements must be rigorously enforced for the GIS offerings, if students
are to receive credit for the course.

We must have on-going control over the quality of GIS Computer Science credit courses.
These courses ultimately affect our reputation as a department, the reputation of our students,
and our ability to attract and educate the best students.

2.1g Summer Courses
In the last five years, there has been a growing demand for offering our first, second and third
year courses during the summer session. Such demand comes primarily from our transfer, second
degree and co-op students who need to complete certain courses before the next winter term,
secondarily from regular Computer Science students who would like to reduce their course load
in the following Winter session, and from non-Computer Science students who are unable to fit
these courses into their Winter session schedule.

For the last two years, the Department has administered the summer courses by itself. The
ability to attract good instructors and teaching assistants (with competitive remunerations), the
ability to control class sizes, and the meticulous work of our administrative staff have made the
summer offerings a significantly better experience for our students. In the future, we plan to
experiment with different duration times to further improve the effectiveness of our summer
offerings.

2.1h Concluding Remarks
In this section, we have so far identified many challenges facing our undergraduate program.
There is no doubt that the dominant challenge is the enrollment pressure we have been facing in
the last five years. Based on the expected enrollment and the desired class sizes mentioned
above, the following table shows our future projections in term of student FTE’s for both core-
CPSC and non-CPSC students.
The above table estimates some key resources that are required for teaching. The table, however, does not include the resources required for the various activities discussed earlier that we wish to undertake (e.g., curriculum revisions, development of new programs, etc.)

The table indicates that we will have to cover 72 lecture sections in each winter term. The current mix of research, faculty members, instructors and sessional lecturers cover 54 sessions. One way to solve this problem is to increase the undergraduate teaching responsibilities of our faculty members. But increasing teaching responsibilities will undoubtedly jeopardize the needed effort for other activities including curriculum revision, development of new programs and research initiatives. Furthermore, given the enormous demand from industry and other universities, we already face a difficult task in hiring qualified people; increasing teaching responsibilities will make it much harder to attract new faculty, or even to retain existing faculty members. Thus, it is imperative that new positions become available.

The table indicates that we will need around 120 Teaching Assistants (TA’s). TA’s are needed to give tutorials and labs for all first- and second-year courses and many third-year courses. Over the last five years, our TA budget has been significantly smaller than our needs. To meet the target of 120 TA’s per year, we need an even larger increase in our TA budget. It is also important to note that increasing the number of TA’s implies a corresponding increase in the number of graduate students. This, in turn, demands the increase in faculty members.

The table indicates that we are in desperate need of classroom and lab space. Of particular difficulty is the lack of classrooms of size 200 or more in the south end of the campus. The current ad-hoc solution is for us to have classrooms and labs spread across campus. This makes it difficult for everybody involved, including the students and the staff.

Technical staff is crucial to the undergraduate program because they create and maintain the lab environment required for almost every course we offer. In addition, they support the faculty in curriculum development. Administrative staff is also crucial to ensure the smooth operation of an undergraduate program of our size. Key administrative activities include registration and waiting list maintenance, advising, prerequisite checks, graduation checks, assessment of institutional and individual transfer credits, student awards, and secretarial support for the courses. Moreover, admission control and enrollment restriction also require additional administrative support. Currently, because of serious under-funding for both technical and administrative staff, a considerable portion of this load has been taken by faculty members, at the

<table>
<thead>
<tr>
<th>Year</th>
<th>CPSC Students</th>
<th>CPSC FTE's</th>
<th>Other FTE's</th>
<th>Instructors</th>
<th>TA’s (@ 12h/w)</th>
<th>200 or more</th>
<th>100</th>
<th>50</th>
<th>Labs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>N/A</td>
<td>1050</td>
<td>1500</td>
<td>17</td>
<td>30</td>
<td>25</td>
<td>0</td>
<td>22</td>
<td>75</td>
</tr>
<tr>
<td>2nd</td>
<td>350</td>
<td>1400</td>
<td>450</td>
<td>17</td>
<td>25</td>
<td>25</td>
<td>0</td>
<td>22</td>
<td>60</td>
</tr>
<tr>
<td>3rd</td>
<td>310</td>
<td>1900</td>
<td>350</td>
<td>19</td>
<td>38</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>4th</td>
<td>300</td>
<td>1300</td>
<td>225</td>
<td>19</td>
<td>26</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>960</td>
<td>5650</td>
<td>2525</td>
<td>72</td>
<td>119</td>
<td>80</td>
<td>30</td>
<td>74</td>
<td>180</td>
</tr>
</tbody>
</table>
expense of other activities such as curriculum development and research. Serious under-funding also lowers the job satisfaction levels of existing staff, to the extent that retention becomes a challenge.

2.2 Structure of the Graduate Learning Environment
2.2a Background
The department currently has 119 graduate students; 82 are M.Sc. and 37 are Ph.D. students. During the last recruiting season, 265 students applied to the graduate program, from which 115 were selected and given admission offers, of which 53 accepted. Just one year ago, the department had 100 graduate students; the sudden increase by 20% has made office space quite tight and has rendered pairing of supervisors to graduate students challenging in some research groups. Aside from research, coursework, and teaching assistantships, students have high participation in most department activities, including social and administrative functions. All departmental committees include at least one graduate student representative. This role is appreciated by the graduate students who feel their involvement is significant and is respected by the department. The Computer Science Graduate Student Association acts as a governing body for graduate students within the department alongside the existing Graduate Affairs Committee and the new Associate Head of Graduate Affairs.

2.2b Graduate Teaching Resources
The department does not presently receive funding for graduate teaching lab space. Several current graduate courses require students to use laboratory facilities. The subject matter of these courses makes it essential for students to have direct access to lab space. As it stands, equipment in research labs is shared with students taking these courses. This situation works well when a graduate course only has four, five, or six students. Over the last few years, however, graduate courses within computer science have grown to twenty, thirty, and sometimes even forty students. When this many students access a particular lab, research within the lab is seriously affected. Researchers within the Integrated Systems Design Lab and the Laboratory for Computational Intelligence seem especially affected; near the end of term, research projects in these labs are hindered and sometimes even halted because of graduate student coursework.

Research laboratory space is funded for the purpose of conducting research. Hence, research is definitely the primary use of a lab. For this reason, some labs are not made available for courses. For example, the image analysis course does not have any lab access for students to run image analysis software.

Two options are foreseeable. Ideally, separate teaching laboratories should be created for graduate courses. These could be made multi-disciplinary to allow for a variety of graduate courses within computer science to have access to equipment. Alternatively, a portion of existing research labs could be paid for by and be made accessible to graduate courses. Both of these options require targeting equipment funds for graduate teaching. In addition, larger enrollment graduate courses would benefit from the help of a teaching assistant. Currently, the department does not budget for teaching assistants in graduate courses. Department budgeting is dominated by the demand for undergraduate teaching which leaves a funding void for graduate teaching. To remain competitive with other large North American research universities, graduate learning, and specifically, graduate teaching laboratories need to be funded.
2.2c Teaching Assistant Pay
Currently, teaching assistants are either paid by salary for a fixed twelve hours of work per week or paid hourly. Some students who are paid by salary are required to work more than twelve hours due to demands of the course. For these students, an overtime hourly pay system would be appreciated so that additional work put in may be compensated.

2.2d Graduate Student Recruiting
Upon completion of a bachelor’s degree in computer science, many students are drawn to work in industry by the current demand and widespread availability of well-paying positions. As a result, recruiting of new graduate students within computer science departments has become competitive. Some graduate students have suggested coordinating visits by current graduate students to other universities, both in Canada and abroad, with information sessions for interested students. A modest honorarium would encourage graduate students to devote their time to recruitment during these visits. Such a system would provide an inexpensive and more personal medium to reach potential graduate students, who may be considering UBC for postgraduate work.

2.3 Learning Environment/Multidisciplinary Efforts
Many areas in Computer Science have strong natural links to other disciplines – Biology, Engineering, Linguistics, Mathematics, Medicine, Music, Physics, Psychology and others. These disciplines apply computer science tools and techniques, and research within them inspires new computational methods.

Increasingly, however, interconnection and awareness across these fields leads to recognition of research challenges with inherent and significant aspects solvable only by expert knowledge spanning traditional boundaries. Computation appears as a theme; recognizing this, the Department of Computer Science has made a goal of providing students with the learning opportunities and training essential to such multidisciplinary ability. Particularly in the context of advanced multidisciplinary studies, our view of the learning environment is a broad one, encompassing courses, involvement in research projects and other academic formats such as discussion groups.

Limited infrastructure and funding for this kind of endeavour has in the past confined us to individual efforts, often supported only by individual start-up funds. We nevertheless report below a number of promising initiatives, which we hope to intensify and complement with additional resources and new hires in the near future.

- **Bioinformatics:** A major effort is currently underway to establish a stimulating and exciting learning environment for bioinformatics. This highly interdisciplinary area attracts and involves students and faculty from Computer Science (CS), Biochemistry, Microbiology, Mathematics, and Statistics. The recently created BETA-Lab (Bioinformatics, and Theoretical & Empirical Algorithmics Laboratory) provides the core infrastructure for bioinformatics activities in the CS department. As we are currently witnessing a strong and increasing interest in this field, we expect that extensions of this laboratory (in terms of space and equipment) will be necessary in the near future. A newly developed graduate course on bioinformatics (currently listed as CPSC 536A and taught by Condon and Hoos) offers students from CS, Biology, Biochemistry, and other departments an introduction to the algorithmic aspects of bioinformatics; this course is
intended to provide one of the starting points towards a bioinformatics program at UBC. Furthermore, frequent guest lectures by CS faculty in courses on bioinformatics and related topics offered by other departments as well as talks by researchers in our department with interests in biology offer frequent opportunities for students interested in bioinformatics to get insights into a range of special topics this field. Finally, collaborative research projects with other UBC departments (in particular, Microbiology and Immunology as well as Biochemistry and Molecular Biology), research institutes (such as the BC Cancer Institute or the Centre for Molecular Medicine and Therapeutics) as well as joint projects with industry (which are currently being established), offer students the chance to get involved in world-class research and development activities in bioinformatics.

- **Cognitive Systems:** A new development at UBC is the creation of the Cognitive Systems program. This program is being developed in response to the tremendous advances achieved over the past few decades by researchers in various areas of cognition. Workers in fields such as artificial intelligence, human perception, and philosophy of mind have found that their work increasingly touches on a common core of issues. As such, there is a need for a program that will cut across traditional departmental lines to focus on these common issues. Cognitive Systems will be a cross-disciplinary program involving four departments: Computer Science, Linguistics, Philosophy, and Psychology. It will provide students with a thorough grounding in the principles and techniques used by intelligent systems (both natural and artificial) to interact with the world around them. It will emphasize the study of existing systems, the design of new ones, and the design of interfaces between different forms of intelligent agents. Three streams will be offered, each co-ordinated by a different department: Computational Intelligence and Design (Computer Science), Language (Linguistics), and Cognition and Brain (Psychology). Each stream will focus on issues most relevant to that department, but will also provide considerable exposure to the problems and approaches developed in the others. All three streams will include courses in Philosophy to help obtain a better understanding of the underlying issues. They will also include the introductory COGS 200 and capstone COGS 400 courses, which will help provide a "big picture" perspective on the ways that intelligent systems can be studied and designed.

- **Computation, Music, and Sound:** As a consequence of relatively recent hires (Hoos, MacLean), existing strengths in sound synthesis and HCI (Pai, Booth), and cooperation with other departments (School of Music, Electrical and Computer Engineering), there is considerable interest and expertise in computer music and sound related research. Students are already working in this area, and are involved in joint interdisciplinary research projects (in particular with the School of Music and the Acoustic Ecology group). We intend to substantially extend these activities and opportunities in the near future, in particular by developing and offering interdisciplinary courses on these topics and by creating a laboratory environment for scientific as well as creative work in sound and computer music.

- **Human-Computer Interaction Focus:** The Computer Science Department is at the core of a multidisciplinary effort to teach students to integrate empirical research on human-
computer interaction with technology, network, and multimedia design for a variety of user activities and contexts. This effort is made possible by a high level of cooperation between research groups within the Faculty of Science and with other faculties such as Applied Science, Arts, Commerce, Education and Medicine. Many of these activities have been coordinated through MAGIC, the Media and Graphics Interdisciplinary Centre, which shares a number of resources with the Imager Computer Graphics Laboratory. In the future much of this activity will be incorporated into the initiatives underway within ICICS, with continuing involvement by members of the department in various projects. These relationships enable UBC students and researchers to more effectively study user behaviours, design prototypes, and test their effectiveness with experimental and observational methods. CS also supports the larger academic community by hosting the PACIFIC seminar on the theoretical basis of HCI and through collaboration with the Faculty of Medicine on Web-based distance learning, namely, MAINPRO-C (highest level) continuing medical education for family physicians throughout BC. We will continue to expand our collaboration with other departments and faculties to build UBC’s leadership in HCI instruction, theory, and application. We are working with Engineering to co-list courses in HCI and graphics to expand their accessibility to students in both disciplines, and hope to extend this collaboration to other departments and faculties (e.g. Psychology and Commerce) for courses in cognitive modeling, decision making and computer-supported collaborative work. In collaboration with these faculties and the new Cognitive Systems Program we will explore the possibility of an interdisciplinary HCI graduate program, perhaps offered under the auspices of MAGIC or ICICS.

• **Microelectronics:** For the past 35 years, the advances in computers have been driven by improvements in integrated circuit technology. During this time, the number of transistors on a chip has grown by more than a factor of a million and the rates at which the components operate has grown by a factor of more than ten thousand. These trends should continue into the beginning of the next decade, and the technologies of microelectronics are likely to be the enabling technologies for future computer technologies such as bio-computation, nano-structures, and quantum computing. UBC enjoys a strong, interdisciplinary, microelectronics research program including faculty and students in Physics, Metals and Materials, Mechanical Engineering (ME), and Electrical and Computer Engineering. This collaboration was recognized last year in a CFI award to create a laboratory for research in the design of high-speed microelectronic circuits, and the successful recruiting of a leading silicon valley entrepreneur to the ECE department. We see this collaboration as the natural base for expansion of UBC research into the areas of computer architecture, embedded systems, and micro-mechanical structures.

• **Physical Interaction:** UBC has the potential to become the North American leader in the emerging field of physical interaction, a new way of perceiving and controlling the results of computation around us. The creation and application of tangible and haptic displays integrates interaction design, operating system and software architecture innovation, the most demanding of mechatronic hardware design, sensory psychophysics, cognition and application disciplines as disparate as Surgery and Music. A growing and
Internally collaborating group in Computer Science (MacLean, Pai), Electrical and Computer Engineering (Fels, Salcudean) and Mechanical Engineering (Hodgson) also has close ties with other universities and enjoys lecturing and course equipment donations from industry collaborators. In CS, project-based course offerings, attracting students from CS, ECE and ME presently include Physical Interface Design & Evaluation (MacLean) and Computational Robotics (Pai). The integration skills of graduates are expected to enjoy high demand in both traditional and IT industries. The diversity in process and skills of any single project or course necessitates cross-department cooperation. This implies mechanisms for students to conduct research, share advisors and implement course projects among multiple departments; and sharing of key infrastructure such as experimentation facilities, mechatronic fabrication, media production studios and teaching laboratories. While their present absence inhibits education and research in this area, we are working to build an environment that supports a free integration of the activities of building, computing and psychological study.

- **Scientific computing**: A major component of scientific computing is interdisciplinary. Both Ascher and Varah have been heavily involved in working and interacting with researchers and graduate students in other departments such as Mathematics, Earth and Ocean Sciences, Mechanical Engineering, Chemical Engineering and more, both at UBC and elsewhere. Many of these efforts have involved the Institute of Applied Mathematics (IAM): in the past eight years, eight IAM PhD students from other departments have graduated under our supervision. A current collaboration between the SCV and the GIF group in Earth and Ocean Sciences provides an excellent opportunity for students and postdocs to be trained in a true multidisciplinary environment. We will continue to be involved in these efforts and to develop new multidisciplinary learning environments involving scientific computing.

- **Social Impact of Computer and Communications Technologies**: By their very nature, these technologies have had and will continue to have a major impact on society in a number of areas including personal privacy, free speech, universal access (the U.S. Digital Divide), intellectual property rights, computer crime and security, education, government, and science. Any attempt to study these issues must include a multidisciplinary approach as they involve such disciplines as Computer Science, Law, Philosophy, Psychology, Political Science, Sociology, Education and Medicine, among others. Early efforts at UBC have involved a graduate course, CPSC 530B, Social Impact of Computers, (Rosenberg) for the past several years, that has included students from Electrical and Computer Engineering, Political Science, and Journalism, in addition to Computer Science. Rosenberg has supervised or co-supervised Ph.D. students in the Law Faculty and in Interdisciplinary Studies with faculty from Anthropology and Sociology, and these efforts are expected to continue.

- **Telehealth**: Telehealth, the use of computing and communication technology to deliver health care, includes a project examining how software for network-based art therapy can
be used as one component in a program for supporting patients with breast cancer. (Booth with PhD student Kate Collie, IISGP, and additional involvement of PhD student Davor Cubranic, BSc student David Lo, and Brian Fisher.)

- **User-Adaptive Systems and Intelligent Learning Environments:** The rapidly increasing availability of sophisticated media on personal computers have substantially enhanced the potential of computers as interactive tools that support a large variety of users on a growing range of tasks. However, designing complex interactive systems that satisfy the needs of individual users from highly heterogeneous user groups is very difficult. Research on User-Adaptive Systems and Intelligent Learning Environments is a multidisciplinary endeavour that brings together researchers and students Artificial Intelligence, Human Computer Interaction, Cognitive Science and Education to understand how new media can be used to design knowledge-based, adaptive systems that provide the user with individualized support for complex learning and reasoning tasks. This endeavour is growing with evolving collaborations among different groups within the CS department, including EGEMS, LCI, and MAGIC/Imager, as well as from Education, Cognitive Science, School of Nursing and the BC Institute for Disease Control. A graduate course that has been taught for two consecutive years by Conati (listed as CPSC 532b, Topics in Artificial Intelligence) currently offers an introduction to the interdisciplinary issues underlying the study and development of user-adaptive systems and intelligent learning environments, and will provide a natural graduate level complement for courses offered in the Cognitive Systems program.

**2.4 Current Initiatives**

These represent considerable diversity and accomplishment. They are ongoing and growing and they indicate a Department committed to serving a wide community of learners, teachers, and developers.

- **Alternative Routes to Computing (ARC)**
  Alternate Routes to Computing (ARC) is a new post-baccalaureate diploma program launched in parallel at both UBC and SFU. The idea grew out of discussions between Maria Klawe and James Lau, Director of IBM-Canada’s Pacific Development Centre in Burnaby. As James says, "ARC is for people who have university degrees in other fields - business, law or biology - who want to enter the information technology industry. In this case, we aren’t waiting for university undergraduates to complete four years of study towards a computer science degree. ARC students come from an older segment of the population, and have a different way of looking at problems. Having varied backgrounds, they will bring a wealth of problem-solving skills and experience to the workplace."

  ARC is a challenging two-year program combining university computer science courses and related work experience. There are four four-month terms of course work separated by an eight-month term of paid work experience in industry. ARC provides additional support in the form of teaching assistants, tutors and mentoring during the first eight months. Those who complete the two-year program receive a Diploma in Computer Science.

- **The CICSR MSS Program**
The Centre for Integrated Computer Systems Research (CICSR) offers a Master Of Software Systems (MSS) degree program. The MSS program is a joint collaboration involving CICSR, Computer Science and Electrical and Computer Engineering.

The program is designed to prepare graduates with degrees in scientific and engineering subjects other than computer science or computer engineering for the specialized area of Software Systems. The MSS program allows students with backgrounds in other technical specialties to upgrade and expand their knowledge to pursue a career in the software industry.

The Program duration is 16 months and is composed of 30 credits of specified courses taken in three semesters (12 months). A four month industry internship is also required.

- **Canada Research Chairs**
Our department developed proposals for Canada Research Chairs (CRC) in three UBC Clusters: Microelectronics and Information Technology, Neuroscience and Cognitive Systems, and Quantum Structures and Information. Alan Mackworth and Nick Pippenger have been awarded Tier I CRC Chairs in Artificial Intelligence and Quantum Structures and Information, respectively.

- **Cooperative Education Program (Co-op)**
The goal of the Co-op Program is to provide undergraduate students with real-life work experience during their course of study. The Co-op Program is a voluntary option offered to students majoring in Computer Science. Students normally apply in their first year of study and are required to have a minimum of a 70% course average as well as excellent interpersonal skills in order to be accepted into the program. Co-op work terms are scheduled in addition to (not as a replacement for) academic terms; the Co-op program adds one full year to the total length of their program over the course of their degree. Students are expected to maintain a minimum average while remaining in the Co-op program. Those students who complete the co-op requirements receive a co-op designation on their transcript upon completion of the program.

The regular schedule for undergraduate students allows for total of five Co-op terms and eight academic terms. Some work terms are done in succession; on average students are placed with a total of three employers. In the past three years, the growth in the Computer Science undergraduate student body led to a 20% increase in the number of applicants to the co-op program every year. As of January 2001, there are 233 undergraduate students in the Computer Science Co-op Program.

In 2000, the Computer Science Co-op Office started a new internship program to help students who are pursuing their second degree in Computer Science and students who have transferred to UBC from another institution gain work experience during their degree. The program consists of placements from eight- to sixteen-months beginning in May 2001 (students will complete a total of four work terms as part of their degree.)

The Co-op office works with employers in BC, Canada and internationally to provide challenging, career-related placements for students. Many of the top IT companies in the lower mainland hire Computer Science Co-op students.

The Science Co-op Programs are administered by a central Co-op office as part of the Faculty of Science. The Computer Science Co-op Program is staffed by a full-time Coordinator and Program Assistant who are part of this unit but have offices in CICSR. This staff works closely with the Computer Science Undergraduate Academic Advisors to administer the program.

**Note:** Funding has been committed from the following sources: CFI ($8.855 million), B.C. ($8.885 million), and UBC ($4.427 million) for the following programs:

- **Advanced Global Communication Systems:** Operating Systems, networks
  Core and Theme: Information Technology
- **Human Communication Technologies:** HCI, interfaces, display, multimedia
  Core and Theme: Information Technology and Cognitive Systems
- **Multi-Agent Information Systems:** Agents, databases, data mining
  Core and Theme: Information Technology and Cognitive Systems
- **Advanced Global Communication Systems:** Operating Systems, networks
  Core and Theme: Information Technology
- **New Models:** quantum computing, computational biology
  Theme: Quantum Computing, Bioinformatics and Functional Genomics, and Biophysics of Macromolecular Assemblies

• **NSERC/Xerox Canada/Sierra Systems Chair in Software Design**

  Gregor Kiczales joined the department in January, 2000, as the newly established NSERC/Xerox Canada/Sierra Systems Industrial Research Chair in Software Design. Gregor's Industrial Research Chair will be supported for five years by Xerox Canada and Sierra Systems Group, a Vancouver-based information technology consulting firm with offices across North America. NSERC provides matching funds.

  Gregor was recruited from Xerox PARC (Palo Alto Research Center) in California. Xerox PARC is one of the world’s leading computer research centers. The first computer designed for an individual - the Alto - was designed at PARC in the 1970’s. PARC also is where icons, pull-down menus, laser printing and Ethernet technology were first developed.

  The Chair has been a catalyst for new activity. Gregor and Gail Murphy established the Software Practices Lab (SPL). Subsequently, Kris De Volder was recruited from Belgium and joined the Department as a new Assistant Professor in January, 2001.

• **SWIFT - Encouraging More Women To Enter Computer Science**

  While the percentage of women in engineering and most sciences at North American universities continues to grow, participation in computer science is bucking the trend. At UBC, for instance, the percentage of women in undergraduate computer science degree programs declined from over 30% in the 1980s to just 16% in 1992, before rebounding to 24%. SWIFT - Supporting Women in Information Technology - was established to identify causes, eliminate barriers and encourage more young women to enter the field.

  SWIFT is a major focus of the NSERC/IBM Chair for Women in Science and Engineering, held by Maria Klawe, established in 1997 by the Natural Sciences and Engineering Research Council of Canada and IBM Canada’s Pacific Development Centre in Burnaby. "There are several SWIFT initiatives under way," says Anne Condon, who joined the department in 1999. "These include surveys of students in several BC school districts aimed at helping us understand why fewer young women are going into information technology." Anne's faculty position was created in support of the NSERC/IBM Chair.
• **WebCT: An Academic And Commercial Success**
Murray Goldberg loves to teach. In fact, in his first year as an Instructor he won a Faculty of Science teaching prize. It was his love of teaching that led to the development and commercialization of WebCT, the world’s most popular Web-based course-preparation system.

"That prize gave me the credibility I needed to get grants related to teaching," he recalls. "I was especially interested in how students’ academic performance would be affected if I put a course on line - and whether the students would accept such a move. So, when I received a grant of $50,000, I set up the study and made a Web-based version of my third-year operating systems course." In an October 11, 2000, news release WebCT announced that its user base has continued to grow dramatically. "Today, more than 70,000 instructors teach over 174,000 WebCT courses at 1,528 colleges and universities in 57 countries...Close to 4.3 million students have seats in an average of 1.9 WebCT courses a year, for a total of almost 8.3 million student accounts during the 2000-2001 academic year."

3. **RESEARCH EXCELLENCE**

The CS Department is in crisis and the University must find a way to solve the problem. If the University does not act, the CS Department will be quickly sapped of its research strength.

- Evaluation of the Department of Computer Science at the University of British Columbia, Interim Report.

3.1 **Current Research Initiatives and Future Research Directions**

- **BETA-LAB/THEORY GROUP**
- Faculty: Anne Condon, Will Evans, Holger Hoos, David Kirkpatrick, and Nick Pippenger
- Adjunct Faculty: Maria Klawe, Alan Mackworth, Raymond Ng, Richard Anstee (Mathematics), Joel Friedman (Mathematics), Tim Menzies (Electrical and Computer Engineering), Jack Snoeyink (University of North Carolina), and Kay Wiese (Technical University of BC)

Our research is focussed on the broad themes of studying computational problems, their formalisation, and complexity, as well as on finding, analysing, and characterising algorithms for solving such problems. The research activities of the group currently range from the study of algorithms for combinatorial and geometric problems from various domains within computer science and other disciplines to quantum and biomolecular models of computation. Algorithms research encompasses both mathematical and empirical approaches, for both sequential and parallel models of computation. Current application foci include bioinformatics and geographic information systems. The recently established BETA-Lab (Bioinformatics, and Empirical, and Theoretical Algorithmics Laboratory) provides vital infrastructure for these research activities, including state-of-the-art computing facilities as well as meeting and working areas. Currently, this research and learning environment is shared by 5 faculty members, 8 adjunct faculty members, 2 post-doctoral fellows, about 10 graduate student members, and a varying number of undergraduate student members.

- **Research Programs: Algorithms and Data Structures for Geometric Objects**
  (including visualization of geometric data, collision detection and motion planning, geometric reconstruction problems, and the interactive simulation of geographic
processes); **Bioinformatics** (including phylogenetic analysis and DNA/RNA strand design) and DNA Computing (study of means for information storage and computations using DNA molecules); **Concrete Complexity Theory** (establishing upper and lower bounds for the resources needed to carry out particular computations in various computational models, including automata, circuits and decision trees); **Empirical Algorithmics** (including the empirical analysis of complex randomised algorithms and systems as well as the experimentally guided design of efficient algorithms); **Facility Location** (involving the optimal location of facilities subject to geometric constraints); **Quantum Computing** (studying the properties of computations in models based on quantum-mechanical rather than classical physical principles).

- **Future Research Directions:** Our general research interests tie naturally into numerous other areas in Computer Science and other disciplines, such as Geography and Management Science (Commerce), Mathematics, Molecular Biology, and Physics. These links provide opportunities for research programs that cross traditional intra-disciplinary and inter-disciplinary boundaries, and we expect that these will spark off new research interests and directions within our group. In the near future, one area in which we will intensify and expand our research activities is bioinformatics (including algorithmic methods for solving problems from molecular biology as well as approaches to biomolecular computing). We are currently experiencing not only a strong and growing interest in this area from students who want to get involved in bioinformatics research, but also from other researchers and research institutions who are seeking our collaboration on bioinformatics projects. Hence, we intend to increasingly engage in multi-disciplinary and cross-institutional research projects in bioinformatics. We are also planning to intensify our activities in the nascent field of Empirical Algorithmics; in particular, we see significant research potential in the combination of empirical and theoretical approaches in algorithm analysis and design. In addition, we forsee growing activity involving the physics of computation, including quantum computation as well as the analysis of hybrid systems.

- **DATABASE RESEARCH**
- **Faculty:** Raymond Ng, George Tsiknis, and Alan Wagner

A database is an electronic filing system - a collection of information stored electronically. Indeed, data, photos, films, videos and other kinds of information can be arranged and accessed much as one would when using a real filing cabinet, or each entry can be electronically linked to all other entries. Whatever the preferred style, more and more companies, hospitals, police forces, governments, and other groups are using databases to store reports, research results, films, publications, pictures, medical records, archival materials and other information.

- **Research Programs:** **Human-Centred, Exploratory Data mining** ("exploratory human-centred data mining," in which computers would do the counting and searching, and human users the abstract thinking and observation), **Outlier Detection** (Outliers are exceptions to general patterns. They’re instances, not patterns), **Image Querying** (designed and developed an expressive query language and interface for image query specification), **Approximate Matching** (the problem of how to estimate the number of matches of a substring in a database), **Parallel Processing Algorithms** (parallel processing algorithms for data-mining and Online Analytical Processing (OLAP))
applications), **Defining Data Models** (define data models that are more general than traditional models).

- **Future Research Directions:** In ongoing and future research initiatives, we intend to pursue such areas as data mining, data warehousing, OLAP and image querying. Apart from hiring new research faculty members on these areas, we should seek to recruit opportunistically on other core database research areas, such as database theory, spatial databases, temporal databases and transaction management.

- **DISTRIBUTED SYSTEMS RESEARCH**
  - **Faculty:** Mike Feeley, Norm Hutchinson, Son Vuong and Alan Wagner
  
  Much of the distributed Systems Group’s (DSG) research targets software that is employed to control and use collections of networked computers. Such computer networks come in all shapes and sizes. And they provide a wide range of services, from air-traffic control to multi-user games, and from parallel processing to electronic commerce. Currently, the focus is operating systems, programming languages, parallel processing, protocol engineering, and networking.

  - **Research Programs:**
    - **Operating Systems** (runs the application software and handles communication between the memory, disk, microprocessor and peripheral devices, such as keyboards, printers and modem),
    - **Programming Languages** (a focus on the writing of programs for distributed computer systems),
    - **Parallel Processing** (involves the use of more than one computer to solve a problem),
    - **Protocol Engineering** (formal methods for the specification, verification implementation, and testing of modern protocols for high-speed networks, multimedia applications and mobile computing),
    - **Networking** (next-generation Internet infrastructure and middleware design issues, including protocols, scheduling, routing, congestion control and multicasting).

  - **Future Research Directions:** The Distributed Systems Research Group envisages addressing the challenge posed by future large, highly dynamic and heterogeneous distributed computing environment. Key research issues include dynamic management of resources, and reconfiguration, to react to changes in workload, to failures, and to other changes. Management applied to providing high-performance services (e.g. web servers) appears to be a closely related problem. In particular, we will investigate **global, scalable distributed systems** to run in the Internet with thousands of clients, particularly to deal with fault tolerance, resource management and consistency issues. Aspects of **data mining for network management** will also be researched, that apply cluster computing to extracting knowledge from large datasets gathered from network management tools. The future research encompasses system area networks, distributed applications, operating systems and middleware design and implementation.

- **E-GEMS**
  - **Faculty:** Maria Klawe, Cristina Conati, and Kellogg Booth

  "E-GEMS" (Electronic Games for Education in Math and Science) is an interdisciplinary group of computer science and education researchers, school teachers and game developers, who are collaborating on the research and development of educational software that uses computer game-like activities to increase student interest and motivation. UBC Computer Science professor Maria Klawe leads the program, and colleagues Cristina Conati and Kellogg Booth are active participants. The E-GEMS group has investigated curriculum, behaviour, interfaces and psychological issues involved in the design and use of computer games for mathematics
education. And it has examined the strategies and materials needed to integrate these games with other forms of classroom learning. E-GEMS has played a leadership role in identifying gender differences in how students interact with computers and software, and the consequences of these differences on learning outcomes.

- **Research Programs:** E-GEMS has created several mathematical computer games for use in classroom learning and in research studies. These include *Counting on Frank* (player’s goal is to help a boy named Henry and his dog Frank win a contest that requires them to guess the number of jellybeans in a jar, developed by Motion Works and published by EA Kids), *Phoenix Quest* (a multimedia, cross-curricular adventure aimed at making mathematics more appealing to 9 to 14-year-old girls), *Super Tangrams* (a multi-level puzzle game designed to study the effects of different design strategies on children’s learning and on their attitude toward complex mathematical concepts), and *Avalanche* (a four-player game to explore collaborative learning in role-playing computer games).

- **Future Research Directions:** Much of E-GEMS current and future work focuses on incorporating student modelling and intelligent coaches in educational computer games.

- **IMAGER GRAPHICS LABORATORY**

  - **Faculty:** Kellogg Booth, Sid Fels, Brian Fisher, Wolfgang Heidrich, Paul Lalonde, Karon MacLean, Ron Rensink, and associated members Michael Meitner, Gail Murphy, Dinesh Pai, Stephen Sheppard, and Jack Snoeyink

The IMAGER Computer Graphics Lab conducts research in computer graphics and human computer interaction. The objectives are to research the generation of synthetic imagery that either provides realistic views of real or virtual environments (with applications to engineering design, marketing and e-commerce, and the entertainment industry), or conveys information in a comprehensible form (with applications to education and decision making). To this end it is necessary to use and expand physical and psychological models, and to acquire new model data by physical measurements and psychological studies.

- **Research Programs:** Geometric Modeling (methods for interactive design of curves and surfaces, in particular splines, polygonal meshes and subdivision surfaces, deformable surfaces), Physical Measurements (measurement of real-world geometry, photometric measurement of materials and light sources), Rendering (photorealistic image synthesis, image-based methods, global illumination, non-photorealistic rendering and technical illustration), Animation (motion control and dynamics), Interactive Rendering and Virtual Reality (hardware-accelerated rendering, immersive environments, augmented reality, haptic handles for interactive applications), Information Visualization (design of visual interfaces to incorporate recent research on human visual cognition, visual attention and visual memory. Application to time and safety critical interactive systems such as "fishtank VR" air traffic control displays), Collaborative Graphical Systems (Shared and distributed display environments for innovative applications such as distance-based art therapy).

- **Future Research Directions:** In addition to ongoing efforts in the existing research projects, there are a number of areas that will require increased attention in the future: establishing a laboratory for image-based measurements of materials and light sources, an increased focus on physically-based animation and dynamics (including motion control), the development of algorithmic tools and geometric representations suitable for
representing complex geometry and deformable surfaces, and a greater emphasis on virtual and augmented reality environments to support activities in a number of other disciplines.

- **INTEGRATED SYSTEMS DESIGN**
  - **Faculty:** Mark Greenstreet and Alan Hu
  With today's rapid advances in computer technology, there are now increasingly powerful integrated hardware/software systems being developed for air-traffic control, factory automation systems, medical instruments, and home appliances, to give only a few examples. They are becoming so complex, in fact, that it's becoming harder (using current testing methods) to guarantee in the design stage that they will work as intended, when built. Not only that, there is increased pressure to decrease development time, most of which involves verification. The Integrated Systems Design (ISD) Lab was set up to create new and fast ways to verify the designs of highly complicated integrated systems.
  - **Research Programs:** *Verification Based on Continuous Models* (verification, combining methods from differential equations with techniques for reasoning about discrete systems), *Verification Based on Discrete Models* (using formal verification techniques to create faster, cheaper and automatic ways to get the "bugs" out of system designs).
  - **Future Research Directions:** As we develop effective tools for analysing continuous and hybrid systems, we expect to employ them first for verification of hardware design and to use the insights that we gain to discover novel, high-performance circuits. The basic problems of reachability analysis are central in many areas of research including robotics, chemical plant control, bio-medical applications, and ecosystem modeling. We expect to explore these connections and develop further interdisciplinary links. Our work in discrete models and verification has already attracted strong industrial interest, and we expect this interest to grow. We are interested in using the techniques that we have developed for hardware verification to software. We have had initial success in the area of verifying performance optimizations made by expert programmers to compiled code for digital signal processors. We see future opportunities in verifying cryptographic algorithms and protocols and for developing aggressive, automatic code optimization methods. We recognize that in the second decade of this century, there is likely to be a major technological shift when we reach the physical or economic limits of silicon based technology. Our research addresses fundamental questions about models of computation, and what can be shown in these models. We cannot predict whether the next computation technology will be based on bio-molecular methods, quantum mechanics, micro-mechanical structures, or some other approach. However, we believe that it is very likely that our research will contribute to the understanding of new models of computation and contribute to their practical deployment.

- **INTERNET COMPUTING LABORATORY**
  - **Faculty:** Norm Hutchinson, Mabo Ito, Son Vuong
  The goal of the Internet Computing Laboratory is two-fold. The first is that of improving the current IP network technology or using the current technology in a more effective way. The second is the development of entirely new paradigms, which attempt to bring realization of the
concept that the network and the computer are becoming indistinguishable, i.e. “the network is the computer”.

- **Research Programs**: Enhancements of Internet Protocols (e.g., TCP, Mobile-IP, BGP, for multimedia applications), Enhancements of Active Networking (where smart packets can carry instructions that can be processed by intelligent nodes – to alter routing and scheduling policies - in a highly dynamic network environment), Web-based Mobile Computing (users can access their home computing environment anywhere anytime via the web browser), Video Streaming (focusing on network issues for efficient video and rich media efficient and reliable delivery over the Internet) Internet Security (protocols and measures against virus and security attacks in the network context), Network Management (the focus is on Web based, wireless and portable solutions for network management).

- **Future Research Directions**: Given the heterogeneous, dynamic nature of the enabling infrastructure and software environments that support current and future Internet applications, it is highly desirable to have a common powerful programming language, paradigm and engine (interpreter) to work with to solve problems of a dynamic and distributed nature, that is elegant, natural and that frees the application developer from concerns of interoperability, compatibility and dynamism imposed by underlying heterogeneous networks and components. Such a language, based on active mobile intelligent objects that can self-spread and create themselves recursively through a logical network, can be viewed as the *network language of the future*. The *active technology* comprising the novel network language and support system will allow the rapid building of infrastructure and middleware for the global seamless communication, coordination and computing for a wide range of Internet applications and pervasive computing. We envisage working on the development of such a novel and powerful language, paradigm and supporting active technology, and to apply them to content-based networking problems and applications, including active web search, intelligent management of mobile and dynamic networks, Internet security management, Internet interactive gaming, and integration of distributed databases for e-commerce applications. We also continue to work on improving the current IP network technology for the next-generation Internet. This includes investigation of research issues related to hardware, protocol and distributed system design for support of media-rich applications requiring QoS guarantees.

- **LABORATORY FOR COMPUTATIONAL INTELLIGENCE (LCI)**
- **Faculty**: Cristina Conati, Holger Hoos, Jim Little, David Lowe, Alan Mackworth, Karon Maclean, Dinesh Pai, David Poole, Richard Rosenberg, and Robert Woodham
- **Associated Faculty**: Uri Ascher, Wolfgang Bibel (University of Darmstadt), Craig Boutilier (University of Toronto), Wolfgang Heidrich, Peter Lawrence, Raymond Ng, Ron Rensink, and Jack Snoeyink (University of North Carolina).

The Laboratory for Computational Intelligence (LCI) supports research in computational reasoning, perception, and robotics. The objectives are to identify the constraints and to define the computations that make intelligent reasoning, action and perception possible. Some of our major research areas include mobile robotics, probabilistic reasoning, reality-based modeling, computer vision, hybrid systems, intelligent computer-aided instruction, collaborative systems,
and remote sensing. The laboratory facilities are shared by 10 faculty members, 3 staff, and an active group of graduate students and post-docs. The Laboratory participates in many of the projects of the Institute for Robotics and Intelligent Systems (IRIS), one of the federal Networks of Centres of Excellence. Some faculty members have affiliations with the B.C. Advanced Systems Institute (ASI) and the UBC Centre for Integrated Computer Systems Research (CICSR). Companies with which faculty have had recent research collaborations include Point Grey Research, Immersion Corporation, MPB Technologies, MacDonald Dettwiler and Associates, International Submarine Engineering, MacMillan Bloedel Research, the Alberta Research Council, Xerox PARC, Ontario Hydro Technologies, AECL, and Inuktun Services.

- **Research Programs:** ACME (Active Measurement Facility, which provides an integrated robotic facility for building reality-based computational models of everyday physical objects), **Robot Partners** (Research on collaborative robot systems), **Constraint-based Agents** (useful in a wide variety of applications in robotics, vision, scheduling, simulation, diagnosis and repair), **Reasoning Under Uncertainty** (probabilistic and decision theoretic reasoning systems with applications in diagnosis, user-adaptive systems and robot control), **User-adaptive Systems and Intelligent Learning Environments** (systems that can effectively adapt their behavior to a user's tasks and relevant features, with focus on applications for computer-based learning, training and collaboration), **Object Recognition** (computer vision systems for identifying and determining the position of previously seen objects and scenes), **Stochastic Search Algorithms** (an algorithmic approach for solving hard combinatorial problems that is based on randomized decisions during search), **Physical User Interfaces** (tangible handles that facilitate expressive and continuous control of computer applications, and a new medium for controlling embedded devices).

- **Future Research Directions:** In addition to its wide range of existing and ongoing research, the Laboratory for Computational Intelligence has identified some areas that need particular strengthening in the future. The areas that are most important for future research development include computational linguistics, machine learning, reasoning, and multi-agent systems. In addition to strengthening these core research areas, LCI intends to develop its capabilities for research at the boundaries with related disciplines, such as cognitive systems (including human-computer interfaces), the integration of computer vision and graphics, real-world robot navigation, and the applications of computational intelligence to education, bioinformatics, databases, and commerce.

- **SCIENTIFIC COMPUTING AND VISUALIZATION (SCV)**
- **Faculty:** Uri Ascher, Paul Carter, Ian Cavers, and Jim Varah

The primary objective of the SCV group is the design, development and implementation of fast, reliable numerical algorithms needed to accurately model a wide variety of physical phenomena. Thus, the group’s research interests encompass both theoretical and applied aspects of scientific computing. Various scientific visualization packages provide a crucial tool in the development and evaluation of these methods. Members of the group have been involved in designing and analyzing general-purpose algorithms for solving differential equations; solving large, sparse linear algebra problems; writing general-purpose mathematical software; considering discrete problems, particularly problems posed by geometric shapes needed to solve larger problems; investigating parallel algorithms and optimization techniques; designing multi-level algorithms; solving inverse problems; and getting more specifically involved in particular application areas.
such as multi-body systems simulation, robotics, data inversion in geophysics, 3D electromagnetic modeling, image reconstruction and computational fluid dynamics.

- **Research Programs: Numerical Linear Algebra** (emphasis is on sparse matrix computations, which has applications in vision, robotics and graphics, as well as inverse problems in many areas), **Constrained Differential Equations** (particularly those arising in virtual reality simulations and electromagnetic simulations, and including problems with discontinuous coefficients), **Inverse Problems** (including computational geometry considerations and solving large, constrained optimization problems), **Multilevel Methods** (with applications in computer vision, fluid dynamics and inverse distributed parameter estimation).

- **Future Research Directions:** All current research programs of the SCV group are rather broadly defined and ongoing. Over the next five years we will continue to pursue the many questions that remain unanswered within their parameters. In particular, additional emphasis will be placed on the various aspects of inverse problems, optimization and virtual reality. To be able to carry out these goals we will need to hire at least one - perhaps two new research faculty members.

- **SOFTWARE PRACTICES LABORATORY**
- **Faculty:** Gregor Kiczales, Gail Murphy, and Kris De Volder,

Real-world software development involves a diverse set of practices. These include design, implementation, debugging, documentation, integration, maintenance, re-engineering, and re-use. The goal of the Software Practices Laboratory (SPL) is to improve current practice by developing methods and technologies to help software developers build better systems--and to do so more effectively. Our work is rooted in deeply understanding both the structure of software systems and the nature of software development as a practice. Research in the laboratory spans the fields of programming languages and software engineering.

- **Research Programs: Aspect-Oriented Programming** (provide programming mechanisms that enable modular implementation of crosscutting program structures), **Structure-Based Analysis** (developing a variety of structure-based tools to help software developers understand and modify systems), **Software Interactions** (working on new, more flexible, approaches to connecting software components), **Assessment Approaches** (various means of assessing emerging software development approaches, such as surveys, case studies, and experiments), **Declarative Meta Programming** (designing, implementing and using hybrid combinations of imperative and declarative programming paradigms to develop more effective programming languages and software development tools).

- **Future Research Directions:** Our future plans include building on our existing strengths and to expand our efforts to address issues that arise earlier in the software lifecycle, such as requirements analysis, and are at a higher-level, such as methods or processes. One specific goal is to integrate our technologies and to engage in more real-world experiments with them.

- **THEORY GROUP:** See BETA-LAB/THEORY GROUP

### 4. LINKS WITH THE COMMUNITY
4.1 Current Efforts

Comprehensive Department Communications Strategy

The ever-broadening nature and scope of computation, intense academic and industry competition for faculty and graduate students, and the enormous popularity of an IT degree with students of a broad range of ability, today create a new environment, unprecedented in both the competitiveness and the interdependency of its players. To achieve excellence we must lure premier faculty from other organizations, seek funding from industry, draw top graduates from other universities and the best secondary students from provincial schools – while our peers and partners do the same.

This department’s current crisis has crystallized a realization that while these activities occur here piecemeal, as an organization we do not explicitly articulate a compatible vision of our identity, contributions and our goals to the diverse audience we need to reach; and we tend to view ourselves as playing in a merely Canadian, rather than a North American, European and Asian field. Therefore, an explicit goal by mid-2001 is to review our situation, objectives, audiences and strategies in this regard, and in the following months to implement the first pieces of a formal communications plan. The Department’s Public Relations Committee has begun this process by consulting with the UBC Office of Public Affairs, and will coordinate with and leverage the Development Office and other School and University resources wherever possible.

The key elements of our germinal plan include an analysis of our competitive advantages and disadvantages, our communications objectives, key messages and target audiences; strategies and timelines for meeting those objectives; and an assessment of the budget required for different strategies. The scope of our plan will depend largely on the resources we identify within and outside the department, and ascertaining them is one objective for the next few months.

4.2 Other Activities, Including the Use of the Robson Square Campus

As part of its growing outreach program, the department is considering establishing several types of informal but ongoing links with local industry and policymakers. While we are in the very early stages of planning this, ideas include:

- Regular "breakfasts" attended by a rotating cast of faculty members and either/both local executives, engineers and policymakers, during which department members can provide mentorship or tutorials about current technical topics in which other attendees have expressed interest. Robson Square would be an excellent venue for this.
- Participation in established meetings of such groups, e.g. the new fora for the Young Presidents Organization (YPO), again to provide technical education at the request of other participants.
- Continuation of past K-12 educational outreach activities, including individual visits by faculty and graduate students to selected schools and clubs, as well as the department’s traditional Open House, where many (>1000) K-12 children come to the department for demonstrations and tutorials during one day.
5. STREAMLINING ACADEMIC AND ADMINISTRATIVE PROCESSES

As part of its longer-term planning process, the Department, together with other administrative units should re-examine its relationship with the Faculty of Science and of Engineering with an eye to determining whether a rejuvenated relationship in its existing faculty is sufficient to ensure the good health of CS, or whether a move to a different faculty is warranted.

- Evaluation of the Department of Computer Science at the University of British Columbia, Interim Report.

5.1 Within the Department
Following the recent external review of the department and the last annual departmental retreat, the department agreed to appoint three Associate Heads one for the Graduate Program, one for the Undergraduate Program and one for Operations and Resources. While not exactly a streamlining of administrative processes, this change is a response to growth and will allow more immediate and effective interaction and planning with students (undergraduate and graduate) and with staff (technical and administrative/clerical).

Enrollment pressure in computer science drove us to develop more efficient and effective methods of course delivery. This led to important innovations in educational technology of which WebCT, the annual “Home Suite Home” CD-ROM (3rd edition 2000) and the “Roadmap to Computing” project are prime examples. At the same time, enrollment pressure has meant that teaching resources have disproportionately been spent to accommodate greater numbers of students in existing courses at the expense of resources to revise and/or develop new course content. Computer Science is a rapidly advancing discipline. As a department, we will fail in our goal to provide leading undergraduate and graduate programs if we do not take a leadership role (and devote the required resources) to develop curriculum content. Development of new course content benefits more than just students in our own courses since students elsewhere benefit from the textbooks written and from the course material prepared. We propose to include (and reward) substantive course content revision and new course development within our departmental model of teaching load.

One experiment that hasn't worked out as well as we would have hoped is the development of a Guided Independent Studies (GIS) version of CPSC 315. While we support the mandate of distance education to provide access to credit courses for non standard students, increasingly enrollment in the GIS version of CPSC 315 is by our regular students. No provision has been made to fund an upgrade to course content in the GIS version of CPSC 315. The regular version of CPSC 315 has undergone considerable revision to the extent that it is now difficult for us to consider the two versions as equivalent. The lesson for all of us is that, in Computer Science, course development is not a one-time activity but something that needs to be done on an ongoing basis.

We are examining course enrollments, program enrollments and teaching loads to achieve smaller class sizes, especially in the 3rd and 4th year, and the ability to offer sections of popular upper level courses in each teaching term (to enhance flexibility in student scheduling).

Currently, the Faculty of Science has no explicit admission to major programs. Students self-select their major typically after first year. Some programs indirectly control admission by restricting enrollment in a key gateway course. This course then serves as the program filter. Computer Science has no such course. Instead, our current enrollment restrictions and waitlists
create a backlog of students waiting to get the required courses they need while taking computer science (or other) elective courses they don't want (or need) to take. This is unfair to students. We are moving to explicitly control admission to all our degree programs.

Currently, the Computer Science Co-op program is administered by the Faculty of Science. An awkward situation arises when students who are admitted to the Computer Science Co-op program by the Faculty of Science fail to meet the departmental enrollment restrictions in place with respect to the courses those students feel they must take in order to be in the co-op program. The department is working with the Faculty of Science Co-op staff to better communicate with students on this point. It is understood that students in the co-op program are subject to the same enrollment restrictions as any other computer science student. The entire financial and administrative structure of co-op programs needs to be re-examined. Of course, there is little incentive to do this while the provincially mandated tuition freeze continues.

5.2 Within the University
It is in the best interests of the university to have research and teaching in information technology permeate the entire institution. At the same time, there are academic and administrative barriers that limit the ability of the Department of Computer Science to be more effective on an institution-wide basis. The department indeed needs to re-examine its relationship to the rest of the university. There are natural linkages with Engineering. There are other linkages, as well. One possibility is to change from a department to a School of Computer Science. Re-structuring as a school would provide greater flexibility in degree programs, facilitate our ability to re-balance academic and professional elements of current program options and allow us to define new innovative program options that link academic computer science with other disciplines in Arts, the physical and life sciences, Engineering and Commerce.

Our department has placed considerable recent emphasis on building an effective long term relationship with the Department of Electrical and Computer Engineering (ECE). We are committed to collaboration and believe that both departments need to succeed in order for either to succeed. In the past year, we each have launched undergraduate options in software engineering. The goal is effective collaboration and complementarity in software engineering and we will be seeking the advice of a joint Board of Studies on how best to achieve this on an on-going basis. We hope cooperation in software engineering will be the basis for further collaboration in other areas of mutual interest in undergraduate and graduate programs.

Our tenured and tenure-track instructors are a tremendous asset to the department and to the university. Instructors who can thrive in a university research environment are rare. Retention of outstanding instructors is made more difficult owing to the lack of a career path (and associated rewards) beyond the level of Senior Instructor. With respect to total career progress increments, a Senior Instructor is equivalent to a career-long Assistant Professor. Second class status is implied in the current collective agreement that speaks to the possibility of a Senior Instructor being promoted to Assistant Professor. Teaching faculty merit a first class designation, career path and associated reward structure.

It is worthwhile to rethink how graduate education is administered and funded. The experience in our department is that pressures of undergraduate enrollment make it difficult to give priority to allocation of resources to equip a lab devoted to graduate teaching. In practice, resources for graduate teaching "piggy back" on existing research infrastructure. This is viable when graduate course enrollments are small and research activity is not adversely disrupted by graduate course teaching. But, this mitigates against large enrollment graduate courses and
against the development of cross-disciplinary graduate level courses that would attract even larger enrollments. The university needs to think about how the goals of the Academic Plan are best instantiated at the graduate level. Distinct budget allocations for graduate teaching and lab equipment are one way to attach priority to this aspect of UBC’s function.

The department has been very pro-active in student advising and has developed what we believe to be a very effective system for student advising. Students, both current UBC students and students inquiring about UBC, still face many frustrations since some issues are dealt with at the department level, others at the faculty level and others at higher levels (like the Registrar’s office). A “one-stop-shopping” approach to advising campus-wide would definitely streamline administrative processes from the point of view of students (provided the information they receive is timely and accurate).

Current processes for interaction with the Registrar’s office and the Student Information System (SIS) create more work at the department level than is warranted. Pre-requisite checks, GPA calculations to implement enrollment restrictions and management of course waitlists are not now handled automatically. For our department, this creates an enormous additional clerical and student advising load especially at the beginning of each term (including the summer term).

Current university policy does not allow one unit to impose enrollment restrictions on students from another unit in programs that require a specific course as part of that program. Thus, agreeing to allow another unit to require a specific course in one of its programs effectively signs a blank cheque since there is no formal mechanism to control enrollment once the initial agreement has been given. This creates uneven standards for student admission in enrollment restricted courses and discourages inter-faculty cooperation in the development of new undergraduate programs. (Current policy has effected our interactions with the Faculty of Applied Science which admits students to programs after first year and which requires several of our high-demand courses in its programs.)

Student misconduct in the form of plagiarism and cheating is endemic in computer science courses. We don't believe that student misconduct necessarily is more prevalent in our discipline than in others. But, it is the case that our discipline provides better tools than most to detect student misconduct when it occurs. Under the University Act, only the President has the right to take disciplinary action in response to student misconduct. Of course, due process must be respected before discipline is imposed. But, the current mechanisms are cumbersome, excessively hierarchical and too time consuming. The net effect is to demoralize faculty and students alike. An effective, experiential learning environment requires scholarly integrity. The university can not pay lip service to this aspect of the learning environment. Administrative processes need to be in place to deal rapidly and effectively with misconduct when it occurs.

6. Optional Elements

6.1 Infrastructure: An Analysis of Space Needs

Much of what is proposed depends on additional space for offices (Faculty staff, graduate students), labs (research and undergraduate), and lecture halls in order to consolidate the department on this side of campus).

The current space inventory for Computer Science is split over three buildings: CICSR/CS (3880 NASM), Klinck (845), and Forest Sciences (238), for a Current Total of 4962 NASM. (NASM = Net Assignable Square Meters. All figures rounded.)
The Computer Science Hiring Plan proposes increasing the number of faculty in the department from 34.5 to 46 over 5 years, with a corresponding increase in graduate students, research labs, and staff. This gives a ratio of 1.33 increase over our current faculty size. However, the department has recently grown significantly without a corresponding space increase and has severe difficulty accommodating its current activities in existing space. Therefore, a more reasonable estimate of our space needs in 5 years would be 1.6 times our current usage for research labs, offices and administrative space.

An alternative approach to estimate the right space ratio for research labs, offices and administrative space is to consider that for several years up to 1999/2000 we had 29 faculty, a shortage of technical and administrative staff and no access to the Forest Science space. So we were operating in 3880+845=4725 NASM with unacceptably crowded conditions. The increase from 29 to 46 faculty gives a faculty ratio of 46/29=1.59. This approach shows that our planned space ratio of 1.6 is about right.

The Hiring Plan sees an increase in undergraduate teaching of 20% over our current enrollments (equivalent to about 100% increase over our 1996 enrollments). The ITBC proposal is to increase our graduates by 50% over our current production, presumably based on an average over the past few years. As our current undergrad lab space is inadequate to cover the most recent increases, we use a figure of 1.4 times current use for undergrad lab space.

Using the figure of 1.6 for research, office and admin space and 1.4 for undergraduate lab space, we arrive at the required space targets by category with the appropriate growth factors, producing a Required Total of 7622, with a Delta (= Required - Current) of 2660.

A key goal for the provision of the new space must be to consolidate the department in one location rather than being spread across multiple buildings. This would require replacing the space given up in Klinck and Forest Sciences. It also should be noted that Computer Science teaches a lab for CPSC 152 in Applied Science space. The size of this lab is about 100 NASM. The on-going lab space needs for CPSC 152, including the expected growth in Electrical and Computer Engineering enrollment, have not be included in the above calculation.

Accordingly, we have shown the increase over the CS space currently in CICSR/CS to be a total increase of 3743, to be accommodated in the ICICS or CICSR/CS buildings.

Finally, the current building has no classroom teaching space and there is a severe shortage of teaching space in the south of campus. Therefore, while new department space is being built, it is essential that this one chance be used to address the needs for department classroom space. The following classroom sizes would be suitable for meeting the department teaching requirements, which would free corresponding space in several other parts of the campus. We teach enough courses in each year of our program to fully book a classroom, and our target class sizes vary somewhat depending upon year. These classrooms should be tiered and wired.

Classroom space requirements for Computer Science in 2005:

- 1 classroom for 300 students (mostly for 1st year courses and plenary lectures)
- 1 classroom for 200 students (mostly for 2nd year courses)
- 2 classrooms for 100 students (mostly for 3rd and 4th year courses)

With suitable coordination with Applied Science and Forestry we could arrange to share these teaching spaces.

With these space resources, the department would finally be able to consolidate its operations in the manner of most other departments on campus while accommodating planned growth. This
is probably our only opportunity to achieve this goal which is essential if we are to play our key role in the future of the university.

7. Final Remark

The Department is very much at a crossroad. If the University chooses to invest in this Department, it is within reach of a top-ten research standing in North America. It is worth the effort. How often is this opportunity available to a University? The quality of the city, the University and the Department are excellent basic ingredients with which to build upon the base of faculty that remain in this Department. Should the University allow the base to crumble, the effort required to completely rebuild is unimaginable. The best recruitment is retention. That must be the University’s first priority.

- Evaluation of the Department of Computer Science at the University of British Columbia, Interim Report.