S.89-74

SIMON FRASER UNIVERSITY

MEMORANDUM

To: Senate

From: L. Salter
Chair, SCAP

Subject: School of Computing Science - Curriculum revisions
Reference: SCAP 89-56

Date: November 16, 1989

Action undertaken by the Senate Committee on Academic Planning/Senate Graduate Studies Committee gives rise to the following motion:

Motion:

"That Senate approve and recommend approval to the Board of Governors as set forth in S.89-74 the curriculum revisions including

New courses	CMPT 730 - 3	Foundations of Programming Languages
	CMPT 740 - 3	Database Systems
	CMPT 760 - 3	Operating Systems
	* CMPT 821 - 3	Robot Vision
	CMPT 841 - 3	Query Processing in Database Systems
	CMPT 826 - 3	Automated Learning and Reasoning
	CMPT 831 - 3	Functional Programming
	CMPT 843 - 3	Principles of Database and Knowledge-Base Systems
	CMPT 814 - 3	Algorithmic Graph Theory
	CMPT 825 - 3	Natural Language Processing
	CMPT 827 - 3	Expert Systems
Deletion of	CMPT 840 - 3	Advanced Topics in Simulation and Modelling
	CMPT 861 - 3	Biomedical Computing
	CMPT 862 - 3	Computer Mapping
	CMPT 892 - 3	Advanced Seminar II
	CMPT 893 - 3	Advanced Seminar III
	CMPT 895 - 5	Directed Reading II

* Registrar's Note:
This is a modernization of the existing course CMPT 821 "Pattern Recognition and Image Processing"

SIMON FRASER UNIVERSITY

School of Computing Science

MEMORANDUM

TO: Graduate Studies Committee,
Faculty of Applied Sciences

FROM: Joseph Peters,
Director of Graduate Studies,
School of Computing Science

RE: Graduate Program Reorganization

DATE: September 21, 1989

The attached document is a proposal for reorganization of the graduate programs in the School of Computing Science. Our M.Sc. program is now 10 years old and our Ph.D. program is 7 years old. Our enrolment in the 89-3 semester is approximately 47 M.Sc. students and 31 Ph.D. students. To date we have graduated approximately 75 M.Sc. students and 8 Ph.D. students. Our graduate programs have not been changed in any major ways since they were introduced. With the benefit of our experience, we feel that it is time for some changes. Since the number of changes is large, the proposed structures of our graduate programs are presented as a new calendar entry followed by explanatory notes. The notes discuss the purposes of the various components of the programs and the ways in which they meet the goals below. It is hoped that this form of presentation will focus attention on the coherence of the proposed organization better than a tedious before and after comparison with the previous organization. The main goals of this reorganization are the following.

1. Reduce the average time for M.Sc. and Ph.D. students to complete degree requirements.
2. Increase the enrollment in our graduate courses.
3. Ensure that both M.Sc. and Ph.D. students obtain reasonable breadth.
4. Simultaneously reduce the trauma and increase the accuracy of the Ph.D. breadth requirement (currently Comprehensive Examinations).
5. Introduce more flexibility of movement between the M.Sc. and Ph.D. programs so that new students can be more accurately placed.
6. Encourage students to start thinking about research topics earlier in their graduate programs.

The major changes are as follows.

1. Six courses have been identified as M.Sc.-level courses stressing broad coverage of a research area. Four of the courses are existing courses that have been renumbered with 700 numbers to distinguish them from 800-level research topics courses. The other two 700-level courses are new courses.
2. M.Sc. students are required to choose at least 3 of their courses from the 700-level offerings.
3. Ph.D. candidates can now satisfy some of their breadth requirement by taking 700-level courses.
4. A course requirement of 9 semester hours has been introduced for Ph.D. candidates.
5. A Research Topics Seminar series has been introduced to encourage students to think about research problems earlier in their graduate programs.

6. The first two semesters of the M.Sc. and Ph.D. programs have been designed together to permit easy transfer of exceptional M.Sc. students into the Ph.D. program. Transfers from the Ph.D. program to M.Sc. program are also possible with little lost time.

7. Six courses that are no longer offered have been eliminated and 11 new courses (including 3 new 700-level courses) which match the current research interests of the faculty have been introduced to replace them. Most of the new courses are currently being offered as Special Topics courses. Five courses have also been renumbered to accommodate other changes. The titles and/or abstracts of 12 existing courses have been modified to be more accurate and descriptive.

Part I. Calendar Description.

For descriptions of Faculty and Areas of Research, Research Facilities, Laboratory for Computer and Communication Research, and Degrees Offered, see the 1989-1990 calendar entry.

Admission Requirements

To qualify for admission to the M.Sc. program, a student must satisfy the general university regulations and must have a Bachelor's degree, or the equivalent, in Computing Science or a related field. To qualify for admission to the Ph.D. program, a student must satisfy the general university regulations and must have a Master's degree, or the equivalent, in Computing Science or a related field.

To qualify for admission to the Ph.D. program before completion of an M.Sc. program, a student must be enrolled for at least two semesters in the M.Sc. program, must complete the M.Sc. Course Work requirement, and must complete the breadth requirements for the Ph.D. program, as described below, within 12 months of first enrollment in the M.Sc. program. Permission is also required from the Computing Science Graduate Program Committee and the Senate Graduate Studies Committee.

For further information, refer to Section 1.3 of the *Graduate General Regulations*.

Degree Requirements - M.Sc. Program

1. Course Work

The minimum course requirement for the M.Sc. degree is 18 semester hours of graduate level course credit. At least 15 of these 18 semester hours must be in Computing Science and at least 9 of these 18 semester hours must be 700-level Computing Science courses. The Graduate Program Committee may require additional undergraduate or graduate courses or a different number of 700-level semester hours in certain circumstances.

2. Research Topics Seminar Series

The Research Topics Seminar Series is presented in the Fall and Spring semesters to acquaint graduate students with the research interests of the faculty. All M.Sc. students are required to attend the Research Topics Seminars presented during their first year of enrollment in the M.Sc. program.

3. Research

(a) The student is required to submit a Research Proposal to the School for approval by the student's Supervisory Committee. The Supervisory Committee consists of the student's Senior Supervisor, at least one other faculty member from the School of Computing Science, and other committee members as appropriate.

(b) The student is required to present a seminar based on his/her thesis research.

(c) The student is required to submit and defend a thesis based on independent work by the student. For regulations governing the composition of M.Sc. Examining Committees and the conduct of M.Sc. Thesis Examinations, see Sections 1.9 and 1.10 of the *Graduate General Regulations*.

Degree Requirements - Ph.D. Program

1. Breadth Requirement

A candidate for the Ph.D. degree must satisfy the Graduate Program Committee that he/she possesses suitable breadth of knowledge in Computing Science. Normally, this requirement is satisfied during the first year of enrolment through a combination of courses and Comprehensive Examinations as described below although exceptions are possible at the discretion of the Graduate Program Committee in consultation with the Ph.D. Breadth Committee. The Ph.D. Breadth Committee is elected annually by and from the faculty of the School of Computing Science.

A Ph.D. candidate must demonstrate breadth in Theoretical Computing Science and four of the following five areas: Artificial Intelligence, Programming Languages and Systems, Database Systems, Computer Design and Organization, Computer Systems. Breadth in an area can be demonstrated by taking the 700-level course in the area or by writing the Comprehensive Examination in the area. In most cases, 700-level courses taken while in the M.Sc. program can be used to satisfy the Ph.D. Breadth Requirement. Breadth in at least two areas must be satisfied by writing Comprehensive Examinations. The Comprehensive Examinations test breadth of knowledge at the M.Sc. level, and are normally written at the end of April during the first year of enrolment. A candidate should inform the Ph.D. Breadth Committee, in writing, no later than the end of January of the first year of enrolment, which examinations he/she intends to write.

The standing of a Ph.D. candidate is reviewed within 12 months of first enrollment by the Ph.D. Breadth Committee. If the Ph.D. Breadth Committee decides that the candidate has not satisfied the breadth requirement, it may ask the candidate to

(i) do additional courses, projects, exams, or other work,
(ii) transfer to the M.Sc. program, or
(iii) withdraw from the University.

The second of these recommendations must also be approved by the Graduate Program Committee and the Senate Graduate Studies Committee.

2. Course Work

In most cases, Ph.D. candidates will be required to take 9 semester hours of Computing Science graduate level course credit during the first year of enrolment in the Ph.D. program. The Graduate Program Committee may waive part or all of this requirement, or may require additional undergraduate or graduate courses in certain circumstances.

3. Research Topics Seminar Series

The Research Topics Seminar Series is presented in the Fall and Spring semesters to acquaint graduate students with the research interests of the faculty. All Ph.D. students are required to attend the Research Topics Seminars presented during their first year of enrollment in the Ph.D. program. Ph.D. students who have satisfied this requirement while in the M.Sc. program are not required to satisfy it again.

4. Research

The major portion of the Ph.D. program consists of original research under the direction of a Supervisory Committee. A Ph.D. Supervisory Committee consists of a Senior Supervisor, at least one other faculty member

from the School of Computing Science, and other committee members as appropriate.

(a) The student is required to pass a Research Area Examination in his/her chosen research area. The student must demonstrate that he/she possesses both the ability, and adequate knowledge of the chosen research area, to pursue and complete original research at an advanced level. The Research Area Examination is conducted by the student's Supervisory Committee and should normally take place within six months after the student has satisfied the Breadth Requirement.

(b) The student is required to submit a written Research Proposal to the School for approval by the student's Supervisory Committee. The Supervisory Committee must be satisfied that the proposed research is appropriate in level and scope for a Ph.D. thesis.

(c) The student is required to present a seminar based on his/her thesis research. This seminar is normally presented a few weeks before the candidate's thesis defence.

(d) The student is required to submit and defend a thesis based on substantial original research. For regulations governing the composition of a Ph.D. Examining Committee and the conduct of Ph.D. Thesis Examinations, see Sections 1.9 and 1.10 of the *Graduate General Regulations*.

For further information and regulations for both the M.Sc. and Ph.D. degrees, refer to the *Graduate General Regulations*.

Graduate Courses

CMPT 601-5 Computing Science Education I
CMPT 602-5 Computing Science Education II

CMPT 710-3 Computational Complexity (formerly CMPT 810)
This course provides a broad view of Theoretical Computing Science with an emphasis on complexity theory. Topics will include a review of formal models of computation, language classes, and basic complexity theory; design and analysis of efficient algorithms; survey of structural complexity including complexity hierarchies, NP-completeness, and oracles; approximation techniques for discrete problems.

CMPT 720-3 Artificial Intelligence (formerly CMPT 820)
Artificial Intelligence brings concepts such as computation, process, sub-procedure, data structure, and debugging to bear upon questions traditionally raised by Psychologists, Linguists, and Philosophers. In this course we will study a representative sample of work in the field. This will include programs which process written English, "see", play games, prove theorems, and solve problems. *Exclusion: CMPT 410.*

CMPT 730-3 Foundations of Programming Languages
This course will cover basic concepts in the area of programming languages. The course will be largely of a theoretical nature and will concentrate on fundamental concepts of lasting importance, rather than topics of current interest.

CMPT 740-3 Database Systems

Review of introductory database concepts; query optimization; concurrency control; reliability and crash recovery; distributed databases; object-oriented databases; knowledge base management systems.

CMPT 750-3 Computer Architecture (formerly CMPT 850)
Parallel processing: SIMD & MIMD systems, pipelining, data flow architecture; microprogramming; control memory minimization, optimization and verification of micro-programs.

CMPT 760-3 Operating Systems
This course will discuss design issues relating to the functionality and performance of modern workstation operating systems, such as methods for sharing memory, file and data objects, and choice of communication protocols. The special needs of high-performance multiprocessor systems and real-time systems will also be addressed.

CMPT 811-3 Distributed Algorithms
This course is an introduction to computation in distributed systems with an emphasis on the design and analysis of distributed algorithms. We will study many of the distributed algorithms that have been proposed for such problems as election, selection, sorting, spanning trees, and routing. Several models of distributed computing will be discussed.

CMPT 812-3 Parallel Computation
This course is a theoretical treatment of parallel complexity theory concentrating on algorithms and models. Topics will include models of parallel computation, parallel complexity hierarchies, basic tools and techniques for the construction of parallel algorithms, and selected advanced topics.

CMPT 813-3 Computational Geometry
This course covers recent developments in discrete, combinatorial, and algorithmic geometry. Emphasis is placed on both developing general geometric techniques and solving specific problems. Open problems and applications will be discussed.

CMPT 814-3 Algorithmic Graph Theory
Algorithm design often stresses universal approaches for general problem instances. If the instances possess a special structure, more efficient algorithms are possible. This course will examine graphs and networks with special structure, such as chordal, interval, and permutation graphs, which allows the development of efficient algorithms for hard computational problems.

CMPT 815-3 Algorithms of Optimization (formerly CMPT 860)
This course will cover a variety of optimization models, that naturally arise in the areas of Management Science and Operations Research, which can be formulated as Mathematical Programming problems.

CMPT 821-3 Robot Vision
This course discusses issues and research results pertinent to robot vision. Topics include depth recovery, for robot navigation, three-dimensional object recognition and scene analysis, model-based approach, parallel vision machines and algorithms, and case study of comtemporary robot vision systems.

CMPT 822-3 Computational Vision
A seminar based on the Artificial Intelligence approach to vision. Computational vision has the goal of discovering the algorithms and heuristics which allow a two-dimensional array of light intensities to be interpreted as a three-dimensional scene. By reading and discussing research papers - starting with the original work on the analysis of

line-drawings, and ending with the most recent work in the field - participants begin to develop a general overview of computational vision, and an understanding of the current research problems.

CMPT 823-3 Formal Topics in Knowledge Representation

This course surveys current research in formal aspects of knowledge representation. Topics covered in the course will centre on various features and characteristics of encodings of knowledge, including incomplete knowledge, nonmonotonic reasoning, inexact and imprecise reasoning, meta-reasoning, etc. Suggested preparation: a course in formal logic and a previous course in artificial intelligence.

CMPT 824-3 Issues in Logic Programming

This course covers the computation model of logic programs, the theory of logic programs, the Prolog language (both pure Prolog and real-life Prolog, with such features as meta-logical predicates, cuts and negation, extra-logical predicates, and pragmatic issues); advanced Prolog programming techniques, such as nondeterministic programming, incomplete data structures, logic grammars, and meta-interpreters; and applications, such as game-playing, equation solving, compilers.

CMPT 825-3 Natural Language Processing

In this course, theoretical and applied issues related to the development of natural language processing systems are examined. Investigations into parsing issues, different computational linguistic formalisms, natural language semantics, and discourse related phenomena will be considered and an actual natural language processor will be developed.

CMPT 826-3 Automated Learning and Reasoning

This course covers topics which are shared both by AI and Cognitive Science. Current AI research papers are examined from the perspective of Cognitive Science, and vice-versa. Topics covered in a given semester will vary, depending on the instructor, but most of the following topics will be addressed in any given semester: Connectionist models of intelligence; "human-like" automated deduction; reasoning by analogy; topics in natural language; automated concept learning; and computational approaches to semantics.

CMPT 827-3 Expert Systems

This course will analyze the Artificial Intelligence theory and practice underlying Expert Systems and survey a number of the pioneering Expert System applications. Topics will include reasoning engines, the rule-based approach, search, model-based representations, constraint propagation, reasoning maintenance, uncertainty, knowledge acquisition, plus practical issues in Expert System development.

CMPT 830-3 Compiler Theory

Precedence, LL(k), LR(k) grammars; SLR(k), LALR(k), L(m)R(k) and LR(k) parsing techniques, transduction grammars; general compiler organization, code generation and optimization; memory allocation for object programs; garbage collection.

CMPT 831-3 Functional Programming

This course will cover functional programming including introduction to a functional programming language, program transformation and verification, implementation of functional programming languages, and other selected topics which may include parallel evaluation of functional programs, analysis of performance, and advanced applications.

CMPT 841-3 Query Processing in Database Systems

Algorithms for data-intensive operations for disk-based, main-memory-based, loosely distributed and tightly coupled databases; analytical and empirical performance studies of database systems.

CMPT 842-3 Concurrency Control in Database Systems
Transactions, recoverability, serializability theory, schedulers, locking, timestamping, optimistic schedulers, multiversion database systems; Recovery, commit protocols, termination protocols; Replicated database systems, quorum-based concurrency control; Distributed snapshot taking, distributed deadlock detection, reliable storage systems; Concurrency control in object-oriented database systems.

CMPT 843-3 Principles of Database and Knowledge-Base Systems
An advanced course on database systems which covers the following topics: semantic data modelling, engineering databases and spatial databases, object-oriented data models and systems, deductive database systems, semantic query optimization, learning and induction in database and knowledge-base systems, and architectures of data-intensive knowledge-base systems.

CMPT 851-3 Fault-Tolerant Computing and Testing
This course will cover concurrent error detection, self-checking networks, design for testability, and built-in self-test. Existing fault-tolerant systems will be studied.

CMPT 852-3 VLSI Systems Design
This course links two fields that traditionally have been considered two separate entities: computer architecture and integrated circuit design. The vehicle used to demonstrate the interaction of layout issues and architectural concepts is metal oxide semiconductor technology.

CMPT 853-3 Computer-Aided Design / Design Automation for Digital Systems (formerly CMPT 863)
Algorithms for logic synthesis and physical CAD/DA. Emphasis on routing, placement, partitioning, and gate-level logic synthesis.

CMPT 891-3 Advanced Seminar
CMPT 894-3 Directed Reading
CMPT 898 M.Sc. Thesis
CMPT 899 Ph.D. Thesis

Special Topics Courses

In any semester a limited number of Special Topics courses may be offered subject to student demand and faculty availability. Details of any Special Topics courses will be posted several months before they are offered.

CMPT 881-3 Special Topics in Theoretical Computing Science
CMPT 882-3 Special Topics in Artificial Intelligence
CMPT 883-3 Special Topics in Programming Languages
CMPT 884-3 Special Topics in Database Systems
CMPT 885-3 Special Topics in Computer Architecture
CMPT 886-3 Special Topics in Operating Systems
CMPT 887-3 Special Topics in Hardware Design

Part II. Explanatory Notes.

1. Admission Requirements

The requirements for admission to the Ph.D. program before completion of the M.Sc. program have been modified to allow students to change programs with little lost time. Also, see the discussion in section 4 below.

2. Research Topics Seminar Series

In this seminar series, each faculty member will give one or two lectures on topics of his/her choice. The seminars will be given in the Fall and Spring semesters each year. All graduate students will be required to attend the seminars presented during the first year of enrolment. The seminar series serves several purposes. Most importantly, it provides new students with an overview of faculty research interests. It should make it easier for students to select thesis topics and could shorten the average completion time of M.Sc. students by more than 4 months. It will also get Ph.D. students thinking about research as early as possible. A second benefit is that it provides a seminar series in the Fall when external speakers are scarce. Attendance at the seminars will be open to all members of the school. This would give faculty members a good opportunity to learn about the research of their colleagues.

3. 700-Level Graduate Courses

Currently, CMPT 810 is the only graduate course that is explicitly intended to cover a broad area at the M.Sc. level. Some of the other regularly offered graduate courses and a few special topics courses are also fairly broad courses accessible to M.Sc. students. The rest of our graduate offerings are advanced research courses. Currently, we offer close to 20 graduate courses each year. In Part I of this document, 6 courses are identified as stressing a broad coverage of a research area. They are given 700 numbers to distinguish them from the 800-level research topics courses.

The general philosophy for these courses is that they should be accessible to the majority of students entering the M.Sc. program. This means that they should not assume too much specialized previous knowledge. It should be possible for students from widely differing backgrounds to obtain reasonably broad knowledge of a research area without being penalized for moderate deficiencies in their backgrounds. The 700-level courses are intended to fit in with some of the following proposals concerning breadth requirements. Therefore, they will have fairly standardized curricula and will have a final exam as a major component.

Three of the six proposed 700-level courses, CMPT 730, CMPT 740, and CMPT 760 are "new" courses. CMPT 810, CMPT 820, and CMPT 850 have been renumbered CMPT 710, CMPT 720, and CMPT 750, respectively. The titles and calendar descriptions of these three courses have been updated, but the content remains the same as previous offerings. Since these are not new courses, no new course forms have been included. CMPT 760 has already been offered three times as CMPT 886, so it is actually an existing course. However, since we want to retain the 886 Special Topics number, it is necessary to propose CMPT 760 as a new course and a new course form is included in the Appendix. Much of the material for CMPT 740 is taken from CMPT 841 which has been offered several times previously. Advanced material has been added to CMPT 841 to replace the material that has been moved to CMPT 740. Although neither course is completely new, new course proposals are included in the Appendix. CMPT 730 is a new course with nothing similar elsewhere in our graduate or undergraduate offerings.

4. Course Work for the M.Sc. Program

The course requirements for the M.Sc. program remain unchanged at 6 courses, with at least 5 from the School of Computing Science. The additional requirement that 3 of these courses be 700-level courses ensures moderate breadth while still allowing considerable room for specialization. Since most students will probably choose one of their 700-level courses to be in their intended research area, the effect is to require breadth in two areas and depth in one area (the research area). The requirement of 700-level courses also permits a smooth transfer of students into the Ph.D. program as discussed below. The current requirement that all M.Sc. students must take CMPT 810 has been removed (but a similar requirement has been added to the Ph.D. program).

5. Ph.D. Breadth Requirement

The current breadth requirement for Ph.D. candidates is to pass a set of five Comprehensive Examinations at the end of the first year of enrolment. There is no course requirement. The examinations are at the senior undergraduate level and have served well as a filter, but they have also caused some problems. Currently, Ph.D. candidates are spending their entire first year attending undergraduate courses and studying for the exams. This is obviously undesirable. A second problem is the psychological effects on the students. Even though the pass rate is over 85%, writing five 3-hour exams in a two week period is a very stressful experience.

The proposed Ph.D. breadth requirement is a combination of 700-level courses and Comprehensive Examinations. A Ph.D. candidate is now required to demonstrate breadth at the M.Sc. level in five of six areas. (One of the previous areas has been split into two.) Theoretical Computing Science is a compulsory area. This is consistent with the current requirement of CMPT 810 in the M.Sc. program. Breadth can be demonstrated by taking a 700-level course or by writing a Comprehensive Exam. At least two Exams must be written, but the choice is the student's. It is expected that most students will choose to write the exams in their areas of strength and take courses to fill in background in weaker areas.

This approach to ensuring breadth is more constructive than the previous set of Comprehensive Examinations. Students can now correct background deficiencies by taking M.Sc.-level courses instead of undergraduate courses, and the exams in the 700-level courses test moderate breadth at the M.Sc. level, instead of the undergraduate level. It is also reasonable to set the Comprehensive Exams at the M.Sc. level because students will be writing the exams in their areas of strength. Giving choice to the students should eliminate much of the fear about Comprehensive Exams without eliminating the filter that faculty members consider to be necessary.

A further advantage of the proposed system is that it provides a good mechanism for determining in which program a student should be. It is often difficult to determine, on the basis of GRE scores, transcripts, and letters of reference, whether a student from an unfamiliar school has the background to attempt a Ph.D. More than two-thirds of our 500 applicants each year are in this category. The proposed system permits a deferral of the decision, when necessary, until we have enough information to make this decision. Since the structures of the M.Sc. and Ph.D. programs are similar for the first two semesters, a student can move from one graduate program to the other with little loss of time or effort.

Another advantage of the course requirement is that Ph.D. students will acquire GPA's. This is important because most scholarships and fellowships are awarded on the basis of GPA's.

6. Course Work for the Ph.D. Program

A requirement of 3 graduate level Computing Science courses has been added to the Ph.D. program. This requirement is redundant for most students since most students will be taking three courses to satisfy the Breadth Requirement. The purpose of the course requirement is to ensure that the current requirement of five Comprehensive Examinations is not weakened by the introduction of 700-level courses. For example, a student could satisfy his/her M.Sc. course requirement by taking only 700-level courses and these courses could then be used to satisfy the Ph.D. Breadth Requirement. Without the Ph.D. course requirement, this student would only have to write two examinations during the first year of enrolment. In contrast, without the Ph.D. course requirement, an M.Sc. student who took 700-level courses in his/her areas of interest and then specialized by taking 800-level courses would be penalized in the Ph.D. program because he/she would be forced to either write Comprehensive Exams in his/her weaker areas or take two or three more courses during the first year of enrolment.

7. Completion Time

The effect of the changes above should be a decrease in the average length of both the M.Sc. and Ph.D. programs by as much as six months. Currently, the average completion time for the M.Sc. program is 7.5 semesters. A typical M.Sc. student takes courses for 3 semesters and then starts to look for a supervisor and a research topic. This often takes at least one more semester, so that the student is just starting on thesis research in the fifth semester. The average M.Sc. student then spends approximately a year finishing a thesis. Under the proposed system, an average M.Sc. student will have completed 5 courses and will have a good idea of possible thesis topics after two semesters. It is then quite reasonable to expect the student to have completed all courses and be doing research by the end of the third semester. This gives a projected average completion time of about 5.5 semesters.

Ph.D. students are now spending their entire first year taking undergraduate courses in preparation for the Comprehensive Examinations. It is usually not until part way through the fourth semester that they start to think about research. Under the proposed system, most students will finish all breadth requirements in two semesters.

8. Ph.D. Research Area Examination and Research Proposal

The Research Area Examination is the same as what is now called the Oral Candidacy Examination. It is an oral examination intended to test depth in a student's research area. The Research Proposal is a written document that is more specific to the student's research topic. It is also a current requirement.

9. Logistics

The calendar descriptions in Part I are set up to give the Graduate Program Committee and the Ph.D. Breadth Committee enough flexibility to deal with unusual cases. Two possible administrative problems and their solutions are the following.

(a) Sometimes it may not be appropriate for a student to take some of the 700-level courses because the courses are similar to courses already taken (either at S.F.U. or elsewhere). An M.Sc. student can simply be asked to satisfy his/her course requirement with other 700-level courses; there's nothing wrong with more breadth. Ph.D. candidates in this situation could be asked to write the final exam in a 700-level course without requiring the term work. They also have the option of writing the Comprehensive Examination in any area.

(b) Occasionally, a student who has previously failed to satisfy the Ph.D. Breadth Requirement may be admitted to the Ph.D. Program. This situation could arise when a promising student is admitted to the Ph.D. program but then moved into the M.Sc. program to correct background deficiencies that were not spotted during the admission process. This situation is clearly an anomaly that would require individual consideration. Typically, such a student would be admitted to the Ph.D. Program with conditions attached.

10. Supervisory and Examining Committees

The compositions of the M.Sc. and Ph.D. Supervisory and Examining Committees have been changed. Currently, an M.Sc. Examining Committee must include a member external to the School of Computing Science and a Ph.D. Examining Committee must include a member from a department or discipline other than the one in which the candidate is working. These restrictions have proved to be awkward in practice and they do not seem to be serving any useful purpose. The requirement that a Ph.D. Supervisory Committee consists of at least three faculty members has been removed for the same reason. The proposed compositions of these committees are nearly identical to those specified in Sections 1.6 and 1.9 of the *Graduate General Regulations*. The only difference is that at least one member of an M.Sc. or Ph.D. Supervisory Committee, besides the Senior Supervisory must be from the School of Computing Science.

11. House-cleaning

The following courses are being deleted for a variety of reasons such as insufficient enrolments and changes in the faculty.

CMPT 840-3 **Advanced Topics in Simulation and Modelling**
CMPT 861-3 **Biomedical Computing**
CMPT 862-3 **Computer Mapping**
CMPT 892-3 **Advanced Seminar II**
CMPT 893-3 **Advanced Seminar III**
CMPT 895-5 **Directed Reading II**

The following courses have been renumbered.

CMPT 710-3 **Computational Complexity** (formerly CMPT 810)
CMPT 720-3 **Artificial Intelligence** (formerly CMPT 820)
CMPT 750-3 **Computer Architecture** (formerly CMPT 850)
CMPT 815-3 **Algorithms of Optimization** (formerly CMPT 860)
CMPT 853-3 **Computer-Aided Design / Design Automation for Digital Systems** (formerly CMPT 863)

The titles and/or calendar abstracts of the following courses have been modified to be more accurate and descriptive. The numbers of some of the courses have also been changed as noted separately above. There is no change in the content or other details of these courses.

CMPT 710-3 **Computational Complexity** (new title, abstract, and number)
CMPT 720-3 **Artificial Intelligence** (new title, abstract, and number)
CMPT 750-3 **Computer Architecture** (new title, abstract, and number)

CMPT 811-3	**Distributed Algorithms** (new abstract)
CMPT 812-3	**Parallel Computation** (new abstract)
CMPT 813-3	**Computational Geometry** (new abstract)
CMPT 815-3	**Algorithms of Optimization** (new number)
CMPT 823-3	**Formal Topics in Knowledge Representation** (new abstract)
CMPT 824-3	**Issues in Logic Programming** (new title and abstract)
CMPT 842-3	**Concurrency Control in Database Systems** (new title and abstract)
CMPT 851-3	**Fault-Tolerant Computing and Testing** (new title and abstract)
CMPT 853-3	**Computer-Aided Design / Design Automation for Digital Systems** (new title and abstract)

12. New Courses

The following new courses are proposed. New Course Proposal forms are included in the Appendix. As previously discussed, three new 700-level courses (730, 740, 760) are being added to provide a full complement of M.Sc.-level courses. Two courses (821, 841) are existing courses that have evolved to the point that we think that a new course proposal should be included for information purposes. Three courses (826, 831, 843) have been offered previously as special topics courses and are proposed for inclusion because they have been popular. The remaining three courses (814, 825, 827) have not been previously offered, but substantial parts of these courses have been taught as sections of other courses.

CMPT 730-3	**Foundations of Programming Languages**
CMPT 740-3	**Database Systems**
CMPT 760-3	**Operating Systems**
CMPT 821-3	**Robot Vision**
CMPT 841-3	**Query Processing in Database Systems**
CMPT 826-3	**Automated Learning and Reasoning**
CMPT 831-3	**Functional Programming**
CMPT 843-3	**Principles of Database and Knowledge-Base Systems**
CMPT 814-3	**Algorithmic Graph Theory**
CMPT 825-3	**Natural Language Processing**
CMPT 827-3	**Expert Systems**

The addition of these courses does not imply an increase in the number of graduate courses being offered each year. They are being added to reflect the current research interests of the faculty more accurately than the previous courses. During the semesters 86-1 through 89-3, 66 graduate courses were offered by the School of Computing Science. Of these 66 courses, only 42 have been courses with descriptive titles. The remaining 24 courses (more than 35%) have been CMPT 88X Special Topics courses. The 88X series of courses was intended to be used for unusual courses that will only be offered once.

Part III. Appendix - New Course Proposals.

CMPT 730-3	Foundations of Programming Languages
CMPT 740-3	Database Systems
CMPT 760-3	Operating Systems
CMPT 814-3	Algorithmic Graph Theory
CMPT 821-3	Robot Vision
CMPT 825-3	Natural Language Processing
CMPT 826-3	Automated Learning and Reasoning
CMPT 827-3	Expert Systems
CMPT 831-3	Functional Programming
CMPT 841-3	Query Processing in Database Systems
CMPT 843-3	Principles of Database and Knowledge-Base Systems

SIMON FRASER UNIVERSITY

MEMORANDUM

To: Joseph G. Peters,
School of Computing Science

From: Sharon Thomas,
Head, Collections Management Office
Library

Subject: NEW COURSE PROPOSALS

Date: October 3, 1989

I have examined the following new course proposals:

CMPT 730	Foundations of Programming Languages
CMPT 740	Database Systems
CMPT 760	Operating Systems
CMPT 814	Algorithmic Graph Theory
CMPT 821	Robot Vision
CMPT 825	Natural Language Processing
CMPT 826	Automated Learning and Reasoning
CMPT 827	Expert Systems
CMPT 831	Functional Programming
CMPT 841	Query Processing in Database Systems
CMPT 843	Principles of Database and Knowledge-Base Systems

Several of these courses have already been taught as special topics offerings and, in any case, all of them fall well within the Library's normal collecting scope.

Our journal collection, although relatively modest, appears to be adequate for immediate purposes. In fact, the one journal requested for CMPT 843, IEEE Transactions on Knowledge and Data Engineering, is already arriving in the Library as part of our comprehensive subscription to IEEE publications.

Nevertheless it is almost inevitable that expansion of the graduate programme into these areas will generate future demands for a more comprehensive collection than we now possess and the Library may well be faced with demands that it cannot meet.

Sharon Thomas

ST:is

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 730

Title: Foundations of Programming Languages

Description: This course covers aspects of programming language syntax and semantics. Formal methods for defining languages and proving programs correct are considered.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 15 When will the course first be offered: 90-1

How often will the course be offered: Annually

JUSTIFICATION:

This course will cover material that every Computing Science Ph.D. student should know. The course will also be of value to M.Sc. students. Similar courses are taught at other universities. Our students have previously covered much of this material through independent study, while preparing for the Ph.D. Comprehensive examinations.

RESOURCES:

Which Faculty member will normally teach the course: F. Warren Burton or Robert Cameron

What are the budgetary implications of mounting the course: None

Are there sufficient Library resources (append details): Yes

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: [signature] Date: 11/10/89

Senate Graduate Studies Committee: [signature] Date: 31 Oct/89

Senate: Date:

11-

Computing Science

Course Outline

CMPT 730: Foundations of Programming Languages

This course will cover basic concepts in the area of programming languages, at the graduate level. The course will be largely of a theoretical nature and will concentrate on fundamental concepts of lasting importance, rather than topics of current interest. The course is intended to cover material that every Ph.D. student in Computing Science should know.

Outline:

1. Programming language syntax, including a review of BNF notation and an introduction to attribute grammers and their use. (Approximately 2 weeks)

2. Axiomatic semantics, including axioms for assignment statements, statement sequences, conditionals and looping constructs. (Approximately 2 weeks)

3. The lambda calculus, including lambda notation, reduction rules, equality, representation of numbers and lists, conditionals and recursion. If time permits, combinators may be covered. (Approximately 3 weeks)

4. Introduction to denotational semantics, including domains, semantic functions, environments and detailed examples. (Approximately 4 weeks)

5. Type theory, including polymorphism and type hierarchies. (Approximately 2 weeks)

Grading Scheme:

Midterm Examination: 40%
Final Examination: 60%

Required Texts:

Initially, the following two texts will be used in the course.

1. Pagan, Frank G., Formal Specification of Programming Languages, Prentice-Hall, 1981.

2. Gordon, Michael J. C., Programming Language Theory and its Implementation, Prentice-Hall, 1988.

Computing Science

Library Requirements

CMPT 730: Foundations of Programming Languages

This course will be taught from the required texts for the most part. However, students may occasionally refer to papers in the following journals.

1. ACM Transactions on Programming Languages and Systems.
2. IEEE Transactions on Software Engineering.
3. ACM Computing Surveys.
4. IEEE Computer.
5. Computer Languages.

The library current subscribes to all of these journals.

Computing Science

Faculty Qualifications

CMPT 730: Foundations of Programming Languages

Both Dr. Burton and Dr. Cameron are actively doing research in the area of programming languages. Both have published a number of papers on a variety of topics in the area of programming languages, including several in leading journals specializing in programming languages (e.g. ACM Translations on Programming Languages and Systems). Both have previously taught other courses in the area of programming languages, including both undergaduate courses on programming languages generally and graduate courses on special topics in programming languages.

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 740

Title: Database Systems

Description: Review of introductory database concepts; Query optimization; Concurrency control; Reliability of crash recovery; Distributed databases; Object-oriented databases, knowledge base management systems.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 15 When will the course first be offered: 90-3

How often will the course be offered: Annually

JUSTIFICATION:

This course is introduced as one of the six foundation courses in the graduate curriculum of this School.

RESOURCES:

Which Faculty member will normally teach the course: Wo-Shun Luk, Jia-Wei Han

What are the budgetary implications of mounting the course: None

Are there sufficient Library resources (append details): Yes

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: Date: 11/10/89

Senate Graduate Studies Committee: Date: 31 Oct/89

Senate: Date:

Course Number: CMPT 740

Title: Database Systems

Description: Review of introductory database concepts; Query optimization; concurrency control; Reliability and crash recovery; Distributed databases; Object-oriented databases, knowledge base management systems.

Justification: This course is introduced as one of the six *foundation* courses in the graduate curriculum of this School.

Course Outline:

1. Review of the following introductory topics:
   - Why Database?
   - Entity-Relationship Model
   - Relational, Network and Hierarchical Models
   - Query Languages (SQL as example)
   - Relational Databases
   - Physical database structures (files, indices, hashing etc.)
2. Relational Query Processing & Optimization
3. Concurrency Control and Database Recovery
4. Distributed Databases
5. Object-oriented Database Systems
6. Knowledge-base Management Systems

Expertise of the Instructor: Dr. Luk has had over ten years of research experience in databases and has published widely in database journals and conference proceedings. He has also served as a referee for numerous database journals and research grant agencies.

Library resources: The journals that are required for this course are already being subscribed to by our library (e.g. ACM Transactions of Database Systems, Information Systems, Journals for Distributed and Parallel Processing and IEEE Transactions on Software Engineering).

Graduate Course Proposal (Appendix)

Competence of Faculty to Teach CMPT-843

Jiawei Han — Ph.D. in Computer Sciences, has taught the course as a special topics course for two years and publishes research in the areas covered by the course.

Wo-shun Luk — Ph.D. in Computer Science, has taught the material related to the course topics and publishes research in the areas covered by the course.

Library Resources

During the terms in which this course was taught as a special topics course, the library acquired textbooks and relevant supplemental information at the instructor's request. In addition, current research papers and class notes are placed on reserve.

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 760

Title: Operating Systems

Description: This course will cover the pragmatic aspects of modern operating systems design and implementation.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 15 When will the course first be offered: 90-1

How often will the course be offered: Annually

JUSTIFICATION:

This course is being introduced as one of the foundation courses in the graduate curriculum.

RESOURCES:

Which Faculty member will normally teach the course: Stella Atkins

What are the budgetary implications of mounting the course: None

Are there sufficient Library resources (append details): Yes

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: Date: 11/10/89

Senate Graduate Studies Committee: Date: 31 Oct/89

Senate: Date:

CMPT 760

GRADUATE COURSE ON OPERATING SYSTEMS DESIGN.

Instructor: Dr. M. Stella Atkins.

This course will cover the pragmatic aspects of modern operating systems design and implementation. No single reference textbook is available to cover this rapidly changing area, so most material will be taken from recent journal and conference papers, which will be handed out on each topic.

The course will include topics chosen from the following areas, depending on the interests of the students.

(1) Review: single machine implementation of UNIX: file system, etc.

(2) The SUN NFS distributed file-sharing system.

(3) Other Distributed operating systems: ANDREW, AMOEBA, V-system and SPRITE.

(4) Distributed data structures -- Linda.

(5) Concurrent Languages for distributed systems -- Synchronising Resources (SR).

(6) Broadcast and multicast primitives and applications.

(7) Multiprocessor systems: Transputers, COSMIC CUBE, HYPERCUBE, DEC's FIREFLY.

(8) Load sharing and performance measurement.

(9) Real-time systems.

(10) Discussion of the newly-released NeXT computer system -- its operating system (based on Carnegie-Mellon University's MACH operating system), its networking capabilities (based on SUN's Network File System), its virtual memory management and its window display mechanisms.

Grading and Assignments

Students will be expected to complete an assignment, such as measuring some aspect of the system, using either our distributed system of SUN-2 workstations running the message-based V-system, or using the SUNs connected to the department's research Ethernet. Students will also be expected to complete a project on a chosen practical subject, probably using either the language SR on our research Ethernet, or the V-system on our Distributed Computing Laboratory Ethernet, or the Next computers, or, if available, the new Transputers. In addition, each student will be expected to make a 30-40 minute presentation on a recent research paper. There will also be an examination contributing 40% of the grade.

Competence of Faculty Member to teach CMPT 760 -- Operating Systems

Dr. Stella Atkins has studied Operating Systems since 1979, when she started her PhD thesis on the role of exception mechanisms in software systems design, working under Dr. David Cheriton at UBC. Dr. Cheriton subsequently moved to Stanford University where he developed the well-known message-based distributed operating system, known as the V-system. Since completing her PhD in 1985, Dr. Atkins has completed many research projects in operating systems design, ranging from low-level modifications to the V-kernel, to high-level concurrency control studies. More recently, since 1988, Dr. Atkins has become involved in the SFU/Triumf positron emission tomography (PET) design group, and has research publications on real-time systems and applications.

Six Masters students have completed research theses under her supervision in operating systems during 1987 to 1989, and two more are currently working on their theses.

Dr. Atkins has taught operating systems to undergraduates and graduates at Simon Fraser University since 1985, and continues to revise the curriculum to reflect recent hardware and software changes.

Library Resources for CMPT 760

No single reference textbook can keep up-to-date with this rapidly changing area, so most material is taken from recent journal and conference papers, which will be handed out on each topic.

The ACM journals are heavily used, and copies are available in the Computer Science Library, as well as the main library. The IEEE journals are available in the main library. IEEE Conference proceedings are usually (but not always) available in the main library; Dr. Atkins subscribes to some of these for more timely access. Other documents are taken from the computer manufacturer's manuals, of which the department has sufficient copies.

One book may be used as a reference: *Distributed Systems: Concepts and Design* by Coulouris and Dollimore. Addison Wesley, publisher 1988.

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 814

Title: Algorithmic Graph Theory.

Description: The course will examine graphs and networks with special structure which allows the development of efficient algorithms for hard computational problems.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 10 When will the course first be offered: 90-3

How often will the course be offered: depending on demand.

JUSTIFICATION:

Our courses on algorithm design stress universal approaches for the most general problem instances. If the instances posses a special structure, more efficient algorithms are possible. Specially structured graphs (chordal, interval, permutation) admit such algorithms and our students should see how to design them.

RESOURCES:

Which Faculty member will normally teach the course: Pavol Hell

What are the budgetary implications of mounting the course: None

Are there sufficient Library resources (append details): Yes

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: Date: 11/10/87

Senate Graduate Studies Committee: Date: 31 Oct/89

Senate: Date:

a) Proposed Course Outline - CMPT 814

ALGORITHMIC GRAPH THEORY

Text: M. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, 1980.

- Algorithms, Data Structures, Complexity.
- Classical optimization problems.
- Perfect graphs, chordal and strongly chordal graphs.
- Algorithms based on linear programming duality.
- Algorithms for interval graphs, p-q trees.
- Permutation graphs and transitively orientable graphs.
- Other algorithms using special structure.

b) Competence

P. Hell is active in research in the area and has supervised one Ph.D. and one M.Sc. in the same area.

c) Library Resources

The library already has a copy of the text. There are a few papers as well, but they are to be found in the standard journals.

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: School of Computing Science Course Number: CMPT 821

Title: Robot Vision

Description: This course discusses issues and research results pertinent to robot vision. Topics include depth recovery for robot navigation, three-dimensional object recognition and scene analysis, model-based approach, parallel vision machines and algorithms, and case study of contemporary robot vision systems.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 10 When will the course first be offered: 90-3

How often will the course be offered: Once every two years.

JUSTIFICATION:

Robot Vision is a young and rapidly growing field. It is necessary to have a graduate course to concentrate on the fundamental issues and contemporary research results in this area. This is a modernization of the existing course CMPT 821, "Pattern Recognition and Image Processing".

RESOURCES:

Which Faculty member will normally teach the course: Dr. Ze-Nian Li

What are the budgetary implications of mounting the course: None

Are there sufficient Library resources (append details): Yes (see attached)

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: Date: 11/10/89

Senate Graduate Studies Committee: Date: 31 Oct/89

Senate: Date:

School of Computing Science

CMPT 821 Robot Vision

Instructor: Dr. Z.N. Li

This seminar course is intended as a survey of issues and research results pertinent to Robot Vision. No textbook is required. A listing of reading materials from recent journals and conference proceedings will be provided.

Tentative Outline

(1) Depth Recovery for Robot Navigation

(2) 3-D Object Recognition and Scene Analysis

(3) Model-based Robot Vision Systems

(4) Parallel Vision Machines and Algorithms

(5) Hierarchical Representations and Systems

(6) Case Study

Recommended Preparation: Knowledge of calculus, computer algorithms, and artificial intelligence

Grading: Class presentation and participation 20%, two small programming assignments 20%, term project 60%.

CMPT 821

Dr. Ze-Nian Li obtained his doctorate degree in the area of computer vision, and is actively engaged in research in robot vision. He has taught this graduate course in U.S. Dr. Li has supervised one M.S. thesis and is presently supervising one Ph.D. and two M.S. theses in this area.

CMPT 821 Library Requirements

The SFU library has all the following journals and conference proceedings needed for this course.

Related Journals:

(1) Artificial Intelligence
(2) Computer Vision, Graphics and Image Processing
(3) IEEE Journal of Robotics and Automation
(4) IEEE Transactions on Pattern Analysis and Machine Intelligence
(5) IEEE Transactions on Systems, Man, and Cybernetics
(6) International Journal of Robotics Research
(7) Pattern Recognition

Related Conference Proceedings:

(1) Proceedings of the IEEE International Conference on Robotics and Automation
(2) Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition
(3) Proceedings of the International Conference on Pattern Recognition

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 825

Title: Natural Language Processing

Description: In this course, theoretical and applied issues related to the development of natural language processing systems are examined. Investigations into parsing issues, different computational linguistic formalisms, natural language semantics, and discourse related phenomena will be considered and an actual natural language processor will be developed.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 10 When will the course first be offered: 91-1

How often will the course be offered: Every second year

JUSTIFICATION:

See next page.

RESOURCES:

Which Faculty member will normally teach the course: Dr. Fred Popowich, Dr. Nick Cercone or Dr. Veronica Dahl

What are the budgetary implications of mounting the course: None

Are there sufficient Library resources (append details): Yes, (see attached)

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: Date: 11/10/89

Senate Graduate Studies Committee: Date: 31 Oct/89

Senate: Date:

School of Computing Science

CMPT 825 Natural Language Processing

Description

In this course, theoretical and applied issues related to the development of natural language processing systems are examined. Investigations into parsing issues, different computational linguistic formalisms, natural language semantics, and discourse related phenomena will be considered and an actual natural language processor will be developed.

Justification

There is currently a great deal of research into natural language processing, particularly as applied to the machine translation of natural language and to the development of natural language interfaces to databases and expert systems. This course is required to familiarize students with the large amount of contemporary research on issues fundamental to natural language processing and to provide hands-on experience with the problems associated with the actual development of natural language processors.

School of Computing Science

CMPT 825 Natural Language Processing

Instructors:
Dr. Fred Popowich
Dr. Nick Cercone
Dr. Veronica Dahl

In this course, theoretical and applied issues related to the development of natural language processing systems are examined. Aside from surveying some traditional approaches to natural language processing, recent research in the field will also be examined. A list of readings from recent journals along with a list of reference texts will be provided. Students will develop a natural language processor during the course.

TENTATIVE OUTLINE:

Linguistic Formalisms for Natural Language Processors:

A survey of specific linguistic formalisms which have formed the basis for natural language processors. This survey will include formalisms selected from but not restricted to the following set.

transformational grammar, lexical functional grammar, government and binding theory, generalized phrase structure grammar, head-driven phrase structure grammar.

Parsing Natural Language:

Examination of different computational tools and parsing strategies for dealing with the formalisms introduced in the first part of the course. Will also consider the use of non-syntactic (e.g. semantic and discourse-related information) both during and after the parse.

GRADING: Will be based on a term project, assignments, and an examination.

RECOMMENDED PREPARATION: Knowledge of artificial intelligence, parsing, and either LISP or Prolog.

CMPT 825 - Library Requirements

The SFU library has copies of all of the following books which will be required as references for the course.

- *Lectures on Contemporary Syntactic Theories* by Peter Sells.
- *Information-based Syntax and Semantics* by Carl Pollard and Ivan Sag.
- *Computational Linguistics* by Ralph Grishman.
- *Prolog and Natural Language Analysis* by Fernando Pereira and Stuart Shieber.
- *Natural Language Understanding* by James Allen.
- *Logic Grammars* by Harvey Abramson and Veronica Dahl.

The *Proceedings from the Annual Meetings of the Association for Computational Linguistics*, the *Proceedings of the International Conferences on Computational Linguistics*, along with the journal *Computational Linguistics* all contain articles which are relevant to the course. Although these publications are not currently available in the SFU library, Drs. Cercone, Dahl and Popowich have copies which can be put on reserve in the library. Copies of other relevant papers will also be put on reserve.

CMPT 825 - Instructors

Dr. Fred Popowich obtained both his M.Sc. (Computing Science) and his Ph.D. (Artificial Intelligence / Cognitive Science) in the area of computational linguistics. He has published papers describing different natural language processors that he has developed. Currently, he is actively involved in several natural language processing projects. These projects are supported by a fellowship by the Advanced Systems Institute of British Columbia and by the Natural Sciences and Engineering Research Council of Canada.

Dr. Veronica Dahl obtained a Doctorat de Specialite en Intelligence Artificiele at the Aix-Marseille II University. She is internationally known for having pioneered the introduction of logic programming into the fields of deductive databases, expert systems and natural language processing. She has also done extensive research on building logic programming tools for processing language, and on producing useful syntheses between linguistic theory and logic grammars.

Competence of Faculty - CMPT 825

Dr. N. Cercone obtained his M.Sc. (Computer & Information Science) and his Ph.D. (Computing Science) in artificial intelligence specializing in computational linguistics and practical AI systems. He is an active researcher in computational linguistics and expert systems and has taught material of this nature for 13 years. He is currently funded by NSERC with operating, intrastructure, strategic, and equipment grants. He is also editor of Computational Intelligence.

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 826

Title: Automated Learning and Reasoning

Description: See attached page

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

At least one graduate or undergraduate AI course, or instructor's permission.

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 10 When will the course first be offered: 90-2

How often will the course be offered: Once every two years.

JUSTIFICATION:

This course has been offered 2 times in the last 2 years, as a special topics course. It covers current research not addressed in our other courses. It had an average enrollment of 9 students, indicating there is demand for the course.

RESOURCES:

Which Faculty member will normally teach the course: Dr. R. Hadley, Dr. F. Popowich

What are the budgetary implications of mounting the course:

Are there sufficient Library resources (append details): Yes

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: Date: 11/10/89

Senate Graduate Studies Committee: Date: 31 Oct/89

Senate: Date:

Course description for CMPT 826

This course covers topics which are shared both by AI and Cognitive Science. Current AI research papers are examined from the perspective of Cognitive Science, and vice-versa. Topics covered in a given semester will vary, depending upon the instructor, but *most* of the following topics will be addressed in any given semester: Connectionist models of intelligence; "human-like" automated deduction; reasoning by analogy; topics in natural language; automated concept learning; and computational approaches to semantics.

CMPT - 826
AUTOMATED LEARNING AND REASONING

In this course we concentrate upon a number of topics which fall into the "intersection area" between AI and Cognitive Science. We will be reading papers on the topics listed below. Not all topics will receive equal attention. Depending upon student interest, we may focus more on some topics than others.

Topics

- The connectionist (or neural net) model of cognition - massive parallel processing, and Boltzman machines for concept learning. The prospects for using neural nets for higher level reasoning.

- The Frame Problem - its relation to Default Reasoning.

- Automated Reasoning - resolution methods vs. natural deduction methods (including semantic tableaus).

- Cognitive approaches towards automated theorem proving, discovering heuristics automatically, and simultaneous forward and backward chaining.

- Reasoning by analogy - adapting plans by analogy.

- Computational theory of human cognition - Pylyshyn.

- Semantics, and Truth. Can these concepts be explained in computational terms? Implications for the computational model of the mind.

- AI approaches to Concept learning and language acquisition - Michalski

- Philosophical aspects of the computational theory of cognition, Searle, relevance of Godel's theorem.

Competence of Faculty to teach CMPT 826

R. F. Hadley - Ph.D. has taught the course as a special topics course for 2 years. Publishes research in the areas covered by the course.

F. Popowich - Ph.D. is in Cognitive Science. Teaches and publishes in Artificial Intelligence, especially Computational Linguistics.

Library Resources

During the terms in which this course was taught as a special topics course, the library acquired relevant supplemental texts at the instructor's request. In addition, current research papers are placed on reserve.

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 827

Title: Expert Systems

Description: See attached page.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 10 When will the course first be offered:

How often will the course be offered: Once every two years.

JUSTIFICATION:

See attached page.

RESOURCES:

Which Faculty member will normally teach the course: Dr. Bill Havens, Dr. Nick Cercone

What are the budgetary implications of mounting the course:

Are there sufficient Library resources (append details): Yes (see attached)

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: Date: 11/10/89

Senate Graduate Studies Committee: Date: 31 Oct/89

Senate: Date:

SCHOOL OF COMPUTING SCIENCE

CMPT 827 Expert Systems

Description

This course will analyse the Artificial Intelligence theory and practice underlying Expert Systems and survey a number of the pioneering Expert System applications. Topics will include reasoning engines, the rule-based approach, search, model-based representations, constraint propagation, reasoning maintenance, uncertainty, knowledge acquisition, plus practical issues in Expert System development. The course assumes a reasonable background in A.I. fundamentals and programming. The course will make use of the facilities in the new Expert Systems Laboratory within the Centre for Systems Science.

Justification

Expert Systems are knowledge-based computer programs which emulate the reasoning abilities of human experts in some limited field of expertise. Expert System technology is rapidly advancing and being successfully applied in a wide variety of real applications. Although within the field of Artificial Intelligence, Expert System methodology has matured into an important discipline in its own right. This course will focus on presenting a coherent survey of Expert System methodology, including both theoretical issues in automated reasoning and practical issues in knowledge engineering, knowledge acquisition and human interfaces.

SCHOOL OF COMPUTING SCIENCE

CMPT 827 Expert Systems

Instructor: Bill Havens

Syllabus

Expert Systems are knowledge-based computer programs which emulate the reasoning abilities of human experts in some limited field of expertise. Expert System technology is rapidly advancing and being successfully applied in a wide variety of real applications. This course will analyse the underlying Artificial Intelligence methodology and survey a number of the pioneering Expert Systems. The course is intended for graduate students with a reasonable background in A.I. fundamentals and programming. The course will make use of the facilities in the new Expert Systems Laboratory within the Centre for Systems Science.

Topics:

- Introduction - What are Expert Systems?; Synthesis and diagnosis tasks, generative paradigm, recognition systems, general / specialized knowledge.

- Historical Development - Pioneering systems and their contributions: MYCIN, PROSPECTOR, R1, INTERNIST, *et cetera.*

- Logical Foundations - Abductive reasoning, inference engines and logic programming, search.

- Expert System Architectures - Rule-based systems, schema and frame representations, forward and backward reasoning, blackboard architectures, explanations, Expert System Shells and programming environments.

- Constraint Propagation - Domain constraints, local / global consistency, k-consistency, practical consistency techniques, relationship to constraint logic programming, use in real-time systems.

- Reasoning Maintenance Systems - Hypothetical worlds, logical support, chronological / dependency bactracking, justification / assumption-based systems, integration with constraint propagation.

- Uncertainty Calculi - Reasoning with uncertainty, abductive reasoning, belief measures.

- Knowledge Engineering - Ontological analysis, knowledge acquisition, learning techniques, explanation capabilities.

- Building Expert Systems - Prototype selection, initial knowledge acquisition, system design, implementation and critical evaluation.

Prerequisite: Introductory Artificial Intelligence (*e.g.* CMPT 410) or consent of the instructor

Text: F. Hayes-Roth *et.al.*, **Building Expert Systems**, Addison-Wesley, 1983.

Grading: a term project, occasional homework assignments and a final examination will be required of each student.

SCHOOL OF COMPUTING SCIENCE

CMPT 827 Expert Systems

Library Requirements

The following books will be useful (but not strictly necessary) as reference materials for this course:

- M. R. Genesereth & N. J. Nilsson (1987) **The Logical Foundations of Artificial Intelligence**, Morgan Kaufmann, Los Altos, CA.

- P. Harmon *et.al.* (1988) **Expert System Tools and Applications,** John Wiley, New York.

- B. G. Buchanan & E. H. ShortLiffe (1984) **Rule-Based Expert Systems,** Addison-Wesley, Reading, Mass.

- E. Charniak & D. McDermott (1984) **Artificial Intelligence**, Addison-Wesley, Reading, Mass.

The following conference proceedings will be useful research materials:

- *Proceedings of the International Joint Conferences on Artificial Intelligence*

- *Proceedings of the biennial conferences of the American Association for Artificial Intelligence*

- *Proceedings of the biennial conferences of the Canadian Society for the Computational Studies of Intelligence*

The following journals will also be useful research materials:

- **IEEE Expert**

- **Computational Intelligence**

Note: All of these references are either already in the Library or are available from the instructor for library reserve.

Dr. N. Cercone obtained his M.Sc. (Computer & Information Science) and his Ph.D. (Computing Science) in artificial intelligence specializing in computational linguistics and practical AI systems. He is an active researcher in computational linguistics and expert systems and has taught material of this nature for 13 years. He is currently funded by NSERC with operating, intrastructure, strategic, and equipment grants. He is also editor of Computational Intelligence.

William S. Havens

Bill Havens is an Associate Professor of Computing and Engineering Science and Director of the SFU Expert Systems Laboratory. He has taught numerous courses in Artificial Intelligence and Expert Systems at both the undergraduate and graduate level. His research program involves developing constraint-based reasoning systems and applying these techniques to industrial Expert Systems' problems.

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 831

Title: Functional Programming

Description: This course will cover functional programming, including programming techniques, implementation methods and other selected topics.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 10 When will the course first be offered: 90-3

How often will the course be offered: Once every two years.

JUSTIFICATION:

Function programming is becoming an increasingly important topic in computing science. This course will prepare some students for M.Sc. or Ph.D. research in functional programming and give other students a solid introduction to this topic.

RESOURCES:

Which Faculty member will normally teach the course: F. Warren Burton

What are the budgetary implications of mounting the course: None

Are there sufficient Library resources (append details): Yes

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: Date: 11/10/89

Senate Graduate Studies Committee: B R Clay Date: 31 Oct/89

Senate: Date:

Computing Science

Course Outline

CMPT 831: Functional Programming

This course will introduce functional programming at the graduate level to students who have only a limited knowledge of the area of functional programming.

Outline:

1. Introduction to a functional programming language (probably Miranda or Haskell). This part of the course will cover the syntax and semantics of a modern functional language. At the some time, functional programming techniques, including use of lazy evaluation and higher order functions will be taught. Approximately half the course will be spent on this material. (Note: Miranda is currently available on several School of Computing machines that graduate students may use.)

2. Program transformation and verification. This part of the course will cover equational reasoning and inductive reasoning about functional programs, including reasoning about infinite data structures. Approximately two weeks will be spent on these topics.

3. Implementation of functional programming languages. Both environment based implementations and combinator based implementations will be considered. Approximately three weeks will be spent on this topic.

4. Other aspects of functional programming. In the remaining portion of the course (a week or two) other topics will be considered. These may include parallel evaluation of functional programs, analysis of performance, advanced applications or other topics.

Grading Scheme:

Midterm Examination: 20%
Class Project: 40%
Final Examination: 40%

Computing Science

Library Requirements

CMPT 831: Functional Programming

Students may occasionally refer to papers in the following journals:

1. ACM Transactions on Programming Languages and Systems.
2. IEEE Transactions on Software Engineering.
3. ACM Computing Surveys.
4. IEEE Computer.
5. Computer Languages.

The library current subscribes to all of these journals.

In addition, students may need to refer to the following books:

1. Peyton Jones, Simon L., The Implementation of Functional Programming Languages, Prentice-Hall, 1987.

2. Field, Anthony J., and Harrison, Peter G., Functional Programming, Addison-Wesley, 1988.

3. Bird, Richard, and Wadler, Philip, Introduction to Functional Programming, Prentice-Hall, 1988.

The library currently has these books.

Computing Science

Faculty Qualifications

CMPT 831: Functional Programming

Dr. Burton is an active researcher in the area of functional programming. He has published over a dozen papers on functional programming, including several in leading journals. He has previously taught other courses on functional programming, both at the graduate level and the undergraduate level, both at Simon Fraser University and the University of Utah.

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 841

Title: Query Processing in Database Systems

Description: See attached page.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 10 When will the course first be offered: 91-1

How often will the course be offered: Once every two years.

JUSTIFICATION:

See attached page.

RESOURCES:

Which Faculty member will normally teach the course: Dr. W. S. Luk

What are the budgetary implications of mounting the course: None

Are there sufficient Library resources (append details): Yes

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: Date: 11/10/89

Senate Graduate Studies Committee: Date: 31 Oct/89

Senate: Date:

Course Number: CMPT 841

Title: Query Processing in Database Systems

Description: Algorithms for data-intensive operations for disk-based, main-memory-based, loosely distributed and tightly coupled databases; Analytical and empirical performance studies of database systems.

Justification: Database Systems, as a research area, has become one of mainstream areas in computing science. Over the years, existing database faculty have changed the focus of their research, new researchers have joined the faculty, old topics have evolved and new topics have sprung up. As a result, many database related courses have been offered as special topic courses. This is the time to re-organize our database curriculum and formally introduce new course titles to prevent proliferation of special topic courses.

Outline:

1. Query processing algorithms for different database systems (e.g. relational databases, deductive databases and object oriented databases)
2. Characteristics of different hardware platforms (e.g. distributed databases, main memory databases and database machines) and their implications in performance of query processing algorithms
3. Performance evaluation criteria: disk accesses, page accesses, computation cost and communications cost.
4. Performance evaluation methodologies: analytic and empirical techniques.

Expertise of the Instructor: Dr. Luk has had over ten years of research experience in databases and has published widely in database journals and conference proceedings. He has also served as a referee for numerous database journals and research grant agencies.

Library resources: The journals that are required for this course are already being subscribed to by our library (e.g. ACM Transactions of Database Systems, Information Systems, Journals for Distributed and Parallel Processing and IEEE Transactions on Software Engineering).

SIMON FRASER UNIVERSITY

NEW GRADUATE COURSE PROPOSAL

CALENDAR INFORMATION:

Department: Computing Science Course Number: 843

Title: Principles of Database and Knowledge-Base Systems.

Description: Deductive Databases, Semantic Data Modeling, Object-Oriented Databases, Expert Database Systems, New D.B. applications in AI and Engineering.

Credit Hours: 3 Vector: 3-0-0 Prerequisite(s) if any:

ENROLLMENT AND SCHEDULING:

Estimated Enrollment: 10 When will the course first be offered: 90-3

How often will the course be offered: Once every two years.

JUSTIFICATION:

New applications of Database and Artificial Intelligence requires an integration of database and AI technology. The fast expansion of research and development in this field proves its necessity and importance.

RESOURCES:

Which Faculty member will normally teach the course: Jiawei Han, Wo-Shun Luk

What are the budgetary implications of mounting the course: None

Are there sufficient Library resources (append details): New journal, IEEE transactions on Knowledge and Data Engineering is preferred. (Vol. 1, 1989).

Appended: a) Outline of the Course

b) An indication of the competence of the Faculty member to give the course.

c) Library resources

Approved: Departmental Graduate Studies Committee: Joseph Peters Date: 22/9/89

Faculty Graduate Studies Committee: Joseph Peters Date: 10/10/89

Faculty: [signature] Date: 11/10/89

Senate Graduate Studies Committee: [signature] Date: 31 Oct/89

Senate: Date:

Graduate Course Proposal

Department: Computing Science Course Number: **CMPT-843**

Title: **Principles of Database and Knowledge-Base Systems**

Course Description:

An advanced course on database systems which covers the following topics: semantic data modeling, engineering databases and spatial databases, object-oriented data models and systems, deductive database systems, semantic query optimization, learning and induction in database and knowledge-base systems, and architectures of data-intensive knowledge-base systems.

Credit Hours: 3 **Prerequisite:** CMPT-740 or equivalent

Justification:

An important recent development in Computing Science is the application of database and artificial intelligence technologies in the development of engineering, information and knowledge-based systems. A good indication of such a trend is the emphasis of such topics in many international conferences, the publications of many new books and dedicated journals. We have offered such a course as a special topics course in AI (CMPT-882) in the academic year of 1987 – 1988 and as a special topics course in Databases (CMPT-884) in the academic year of 1988 – 1989. Students love the contents of the course and feel the importance of the field. We would like to promote the course to a graduate-level regular course to enrich our curriculum.

Faculty members teaching the course: Jiawei Han and Wo-shun Luk

Both have teaching experience in the area, have publications in journals and conference proceedings, and are referees for some research journals and funding agencies in the area.

No new budget is required to set up the course.

Outline of the course:

(1) Semantic data modeling,
(2) Engineering databases and spatial databases,
(3) Object-oriented data models and systems,
(4) Deductive database systems,
(5) Compilation and optimization of recursive rules and queries
(6) Semantic query optimization
(7) Learning and induction in database and knowledge-base systems, and
(8) Architectures of data-intensive knowledge-base systems.