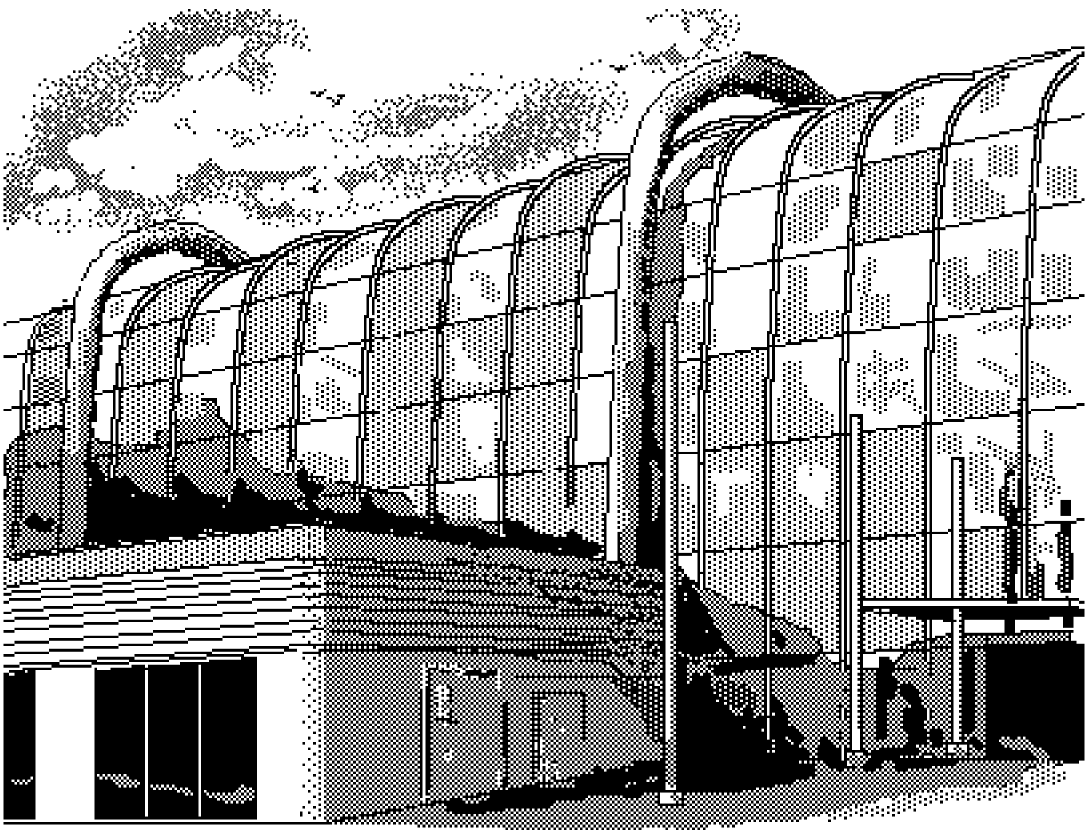
COMPUTER SCIENCE AT WATERLOO

A HISTORY TO
CELEBRATE 25 YEARS

by Peter J. Ponzo



The William G. Davis Centre for Computer Research
University of Waterloo

COMPUTER SCIENCE at WATERLOO

A History to Celebrate 25 Years : 1967 - 1992

Table of Contents

Published by
Computer Science Department
University of Waterloo
Waterloo, Ontario, CANADA
N2L 3G1

Telephone: (519) 885-1211, extension 4464

Printed by Graphic Services, University of Waterloo,
Waterloo, Ontario, CANADA

Acknowledgement

As the 25th Anniversary of the Computer Science Department approached, we formed a committee to determine ways in which we could mark the occasion. At the first meeting it was suggested we try to compile a brief history (perhaps 10 pages or so) to capture some statistical details as well as a few anecdotes.

We approached Peter Ponzo of our Applied Mathematics Department to persuade him to be the author of this document. We selected Peter because he has been at the University of Waterloo since 1957, and has always had an interest in computing and its development at Waterloo. Peter accepted the challenge without hesitation.

With only four months to complete the project, we agreed Peter would ask for input as broadly as possible, and he would try to integrate all of it in a consistent anecdotal style (with cartoons!). We recognized that in such a short time he would probably not hear from everyone, but that he would proceed with whatever was provided. The result is this short history (150 pages!) which in fact captures much more information than we would have thought possible in such a limited time.

We wish to thank Peter for what has turned out to be four months of dedicated effort, all on a volunteer basis.

J. Wesley Graham
Chairman,
25th Anniversary Committee

Foreword

Dear Reader:

In late 1991, Wes Graham, acting as chairman of the 25th anniversary committee of the Department of Computer Science, asked me to write a *History of Computer Science* to celebrate the anniversary (the celebrations to be held in June, 1992). Although I was neither a member of the department nor intimately involved with developments in CS, I listened to his arguments of *unbiassed writing* (Wes is so naive) and *you've been around forever* and *you're the only one stupid enough to agree* [a] ... and I agreed (to the last, if not the first).

I point out, at the beginning, that this *history* is less an accurate disposition of events and more a collection of stories (sometimes vaguely recalled from fading memory). Perhaps there are things mentioned which may offend. I hope not. There are undoubtedly biassed accounts of events (in spite of Wes' expectations). I hope not too biassed. Finally, although I would like to admit to being the source of all errors in this *Computer Science History* (CSH) I will only admit to putting any such errors into print. Further, I made no conscious decision concerning what to include and what to omit - I simply included those relevant and/or interesting events/stories/data about which I could (with relative ease) collect information (including some things not directly concerned with the Computer Science Department at Waterloo).

In December, 1991, I spoke at some length with Ralph Stanton.[b] After an inordinate amount of sherry he related the early part of the *First Decade* which, I feel, is a necessary preamble to the 25 years since 1967. At this point I must admit to a tendency to elaborate (massage? dramatize?) in order to avoid a dry history too filled with facts chronologically ordered. Errors in *dates*, *time of day*, *who said what to whom?* and *what was he wearing at the time?* are inevitable results of this tendency.

Subsequent anecdotal *history* is a consequence of discussions and/or correspondence and/or e-mail with a host of staff and students, a gaggle of

---

[a] just kidding, Wes.

[b] Ralph also browsed thru the early chapters and made verbal note of my short-term memory. (PJP)

faculty, a pride of deans, including Wes Graham, Don Cowan, Barry Wills, Jim Dodd, Frank Tompa, John Brzozowski, Keith Geddes, Greg Fee, Benton Leong, Ian Allen, Bev Marshman, Doug Lawson, Eric Mackie, George Hill, Gary Embro, Bob Zarnke, Richard Bartels, Kelly Booth, Erna Unrau, Jack Kalbfleisch, Doug Wright, Bill Forbes, Alan George, Paul Dirksen, Jim Kalbfleisch, Dave Sprott, Eric Manning, Bruce Simpson, Jerry Bolce, Peter Roe, ... (this list in no particular order).

Finally, I mention that the Prologue, *from Babbage to Turing*, was inserted late in the project to provide an interesting background; sometimes sad, sometimes amusing. (It it even more amusing to imagine this a *History of Computing on Planet Earth*, beginning with Charles Babbage, passing through Alan Turing, Ralph Stanton, Wes Graham and Don Cowan, and ending with Computer Science at Waterloo.)

Thank yous:

A very very special and affectionate *thank you* to Wendy Rush who went out of her way to provide me with machine-readable data.

Thanks, too, to Sandra Ward (for information on WATFOR and WATFIV), to Redina Caracaz (for a complete set of Math Faculty Council minutes starting with the *department* of mathematics, before it became a Faculty), to Jane Britton (for UW archival material), to George Hill (for helping me get this CSH thru' Graphics Services by the deadline) and to Peter Ponzo (for the marvelous cartoons).

I have enjoyed the task and hope you find something of interest.

Peter James Ponzo,
Applied Math Department,
University of Waterloo,
May, 1992.

P.S. This CSH® was prepared on a Macintosh® microcomputer using the word processor Microsoft® Word® and the MacPaint® program embedded in Hypercard®.

Prologue

"Computers" from Babbage to Turing

For 28 years, during the 19th century, an English school teacher, William Shanks, spent his evenings computing the first 707 digits of the number π ... and made an error in the 528th place. It took years - and he made an error. He had used multiplication tables and tables of logarithms, yet each table had taken years to generate - and were filled with errors. The people who generated the tables, and indeed Shanks himself, were "computers" in the original sense of the word.

Speed and *reliability* were the bane of "computers" ... and would be for another century.

******

In the seventeenth century, mechanical calculators for performing addition/subtraction (and multiplication/division ... with some help from the operator) had been devised by the German astronomer and mathematician Wilhelm Schickard. The French mathematician Blaise Pascal later built such a device and later still, in 1673, the German mathematician Wilhelm Leibniz (who, with Isaac Newton, co-invented calculus) built a mechanical *multiplier*.

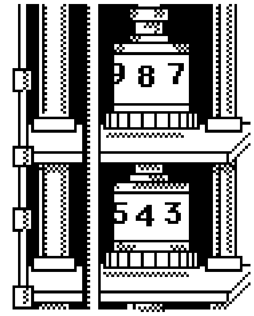
«Leibniz was to write of Pascal's device *"... it facilitates only additions and subtractions, the difficulty of which is not very great ..."* »

******

Charles Babbage

In the early nineteenth century the eccentric Cambridge mathematician Charles Babbage (1791-1871) employed two human *computers* to generate mathematical tables for the Astronomical Society, checking the work of one against the other in order to identify errors.

About 1820, frustrated by the time necessary, and the inevitable errors, he conceived of a "Differential Engine" which would compute the values of functions for equally spaced values of the independent variable, using the *method of differences*.



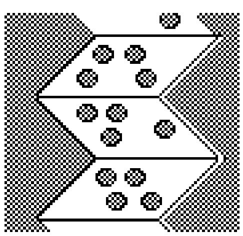
Portion of Babbage's Difference Engine

It was a mechanical device, replete with wheels, gears and levers - and a crank which would successively generate the function values *automatically* (without human intervention - except to turn the crank) once the function definition had been set into the machine. Errors in navigation tables had been the cause of ships running aground, astronomical observations required accurate tables, so Babbage's Differential Engine was of great importance ... and the British government funded the project.

By 1833, 10 years after the project had begun, there was little to show for their investment so the government withdrew the funding and all work on this "Engine" ended. Undeterred, Babbage designed a more ambitious, a more versatile "Analytical Engine".

It would consist of three parts:

- the "store" where numbers were stored, or remembered.
- the "mill" where arithmetical operations on numbers taken from the store would be performed.
- the "sequence mechanisms" which would select the proper numbers from the store and instruct the mill to perform the proper operations.



It was, indeed, an *automatic, general purpose, programmable computer* complete with CPU and memory and input/output and programs and conditional (IF ... THEN) branching and micro-programming - and it printed its answers. Further, it took its instructions from punched cards.

«*Punched cards was an idea Babbage extracted from the textile industry where Joseph M. Jacquard, in 1805, used punched cards to control which threads of warp were lifted above and which below the shuttle. In 1886, a U.S. statistician, Herman Hollerith, would also use punched cards in statistical and accounting machines, to expedite the taking of the U.S. census. (It had taken seven years to perform the clerical work for the 1880 census.) For Babbage, "Operation" cards identified the operation to be performed, and "Variable" cards located the operand in the "store".*»

******

Charles Babbage was a well known scientific and political figure and held parties ... often. His guests might include Darwin or Dickens or Longfellow.

«*Apparently Babbage would read Tennyson (among other things). When he read the lines "Every moment dies a man/Every moment one is born" he wrote to Tennyson to inform him that "... this calculation would keep the world's population in a state if perpetual equipoise." Babbage suggested: "Every moment dies a man/And one and a sixteenth is* *born."* »

His guests would come, be entertained ... and ignore his partially constructed "Engine". In 1833, one guest alone understood: the "Princess of Parallelograms", seventeen year old daughter of Lord Byron, destined to become the Countess of Lovelace, Babbage's "Enchantress of Numbers", the world's first computer programmer, the beautiful and brilliant Augusta Ada. We owe to her a detailed account of both the hardware and software for the "Engine".[a]

Countess of Lovelace

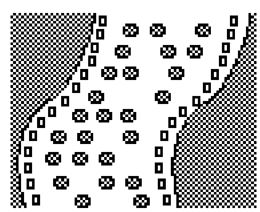
Alas, lack of funds prevented the completion of the Analytical Engine.

On October 18, 1871, Babbage died, disappointed, frustrated, embittered.

******

During the years 1935 - 1945, the the word "computer" took on its modern meaning of a *device* which computes, rather than a *person* which computes.

In the early 30's, in Berlin, a Civil engineering student named Konrad Zuse, in typical student fashion (*sic*), was "too lazy" to perform the endless calculations necessary for his studies.[b] He designed a machine based upon the *binary* number system using telephone relay/switches as on-off devices. (Earlier machines had been *decimal*, incorporating the digits 0 through 9 on toothed wheels ... and for the next ten years computers would be *decimal*.) It took four switches to add 1+1, in binary. His machine filled a small room (his parents' living room!). By 1939 he was the world's leading computer designer.



Then came World War II and he received unlimited funding from the German military. Yet, to overcome war-time shortages, his "programs" were on punched tape made from discarded movie film. By 1941 he had built Babbage's dream machine: an *automatic, programmable, general purpose* computer, using *binary* arithmetic and, unfortunately, electromechanical switches. They were slow, taking up to five seconds to perform a simple multiplication.

---

[a] In the 1970's, the U.S. Defense dept. contracted for a programming language to permit all of its computers to "talk" to each other. It was named ADA.

[b] In a recent TV interview, Zuse says: "I was too lazy ... so I invented the computer."

A friend visiting his laboratory suggested using a *switch* from the new electronics industry: vacuum tubes. As his successors in Computer Science would do, to the present day, Zuse made up a proposal for government funding: it would be a two-year project and would result in a computer 1000 times faster than the earlier version.

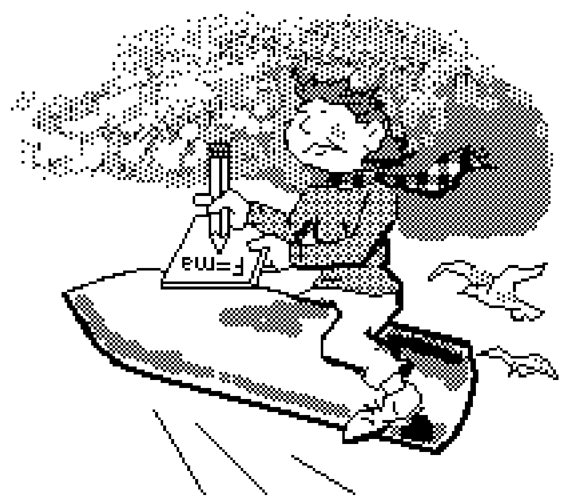
Two years? Hitler knew the war would be won in less than two years - so the project wasn't funded. It would be many more years before the world knew of these developments in Germany.

«*Zuse managed to sneak his latest model (the Z4) out of Germany, accompanied by rocket scientist Wernher von Braun. He later started a computer company that was bought out by Siemens.*»

******

By 1939, the Bell Telephone Laboratories (using a design of a Bell mathematician, George Stibitz, who also invented *floating point arithmetic*), had also built a *binary* calculator which they called a "Complex Computer" to perform the troublesome arithmetic necessary for dealing with complex numbers. As one might guess, it also used telephone relay/switches. A 1940 demonstration, in New York, had the user sending problems from New Hampshire via teletype.

When the U.S. entered the war, one of the most distressing problems was the time necessary to supply the latest weaponry. There was a drastic shortage of human *computers* to calculate the trajectory of shells. "Firing tables", incorporating the effects of wind, temperature, angle of elevation, etc., came from test firings at the Aberdeen Proving Grounds in Maryland.



Hosts of female *computers* were employed; it took 30-40 minutes, with a desk calculator, to compute a single trajectory. It would take four years for a single *computer* to compile one firing table, with its 1800 trajectories.

Howard H. Aiken of Harvard University, inspired by Babbage's ideas, worked with International Business Machines to build, in 1944, the Harvard-IBM Automatic Sequence-Controlled Calculator (ASCC, or Harvard Mark

I). It was a *decimal* machine with 72 storage registers handling 23 decimal digits it performed additions in 1/3 second, multiplications in 6 seconds and worked 24 hours a day solving urgent military-computation problems ... but it contained gears and wheels and relay/switches and failed to take full advantage of the speed inherent in using vacuum tubes. Four years later, on January 27, 1948, IBM was to demonstrate the Selective Sequence-Controlled Electronic Calculator (SSEC), with 13,500 tubes ... and 21,400 electromechanical relays![c]

The problem of *speed* was taken up by John W. Mauchly, a Physicist, and J. Presper Eckert, a young Engineer, during the years from 1942 to 1946. Both were at the Moore School of Electrical Engineering, University of Pennsylvania. With Army funding, they designed a fully electronic computer with some 18,000 vacuum tubes. In spite of the unreliability of vacuum tubes (they burned out - rapidly!) and a prediction that their "computer" would break down every five seconds (the old *reliability* problem), the Army was desperate. The machine was built, filling a 50 foot by 30 foot room with tens of thousands of electrical components in addition to the 18,000 tubes, and some half million soldered joints ... and they called it an *Electronic Numerical Integrator And Computor*.

They demonstrated ENIAC's awesome computing power (with its panel of flashing lights specially wired to impress the cameras of the world) by calculating the trajectory of a shell which would take 30 seconds to reach its target - and they did the calculation in 20 seconds.

Alas, it was not finished until 1946, after the war had ended.

"Programming" required setting some 6000 switches and incorporating hundreds of cables, effectively rebuilding the machine for each problem. Unfortunately, ENIAC was required for the war effort so that this early design had to continue unabated. Before it was put into service, Eckert and Mauchly realized the need for *stored program computers* and left the University of Pennsylvania to start the first commercial enterprise to build computers: *the Eckert-Mauchly Computer Corporation.* They had in mind a *stored programmed* Electronic Discrete Variable Automatic Computer, the EDVAC.

John von Neumann, a Hungarian who came to Princeton in 1930 (perhaps the most distinguished mathematician of the day), learned of ENIAC/EDVAC, visited Eckert and Mauchly and wrote a 101 page *First Draft of a Report on EDVAC*, laying the theoretical groundwork for how

---

[c] a 1981 book by R. Moreau (IBM-France) claims the SSEC to be the world's first *computer*. Earlier machines, including the ENIAC, he calls *universal calculators*.

such a machine should be built.

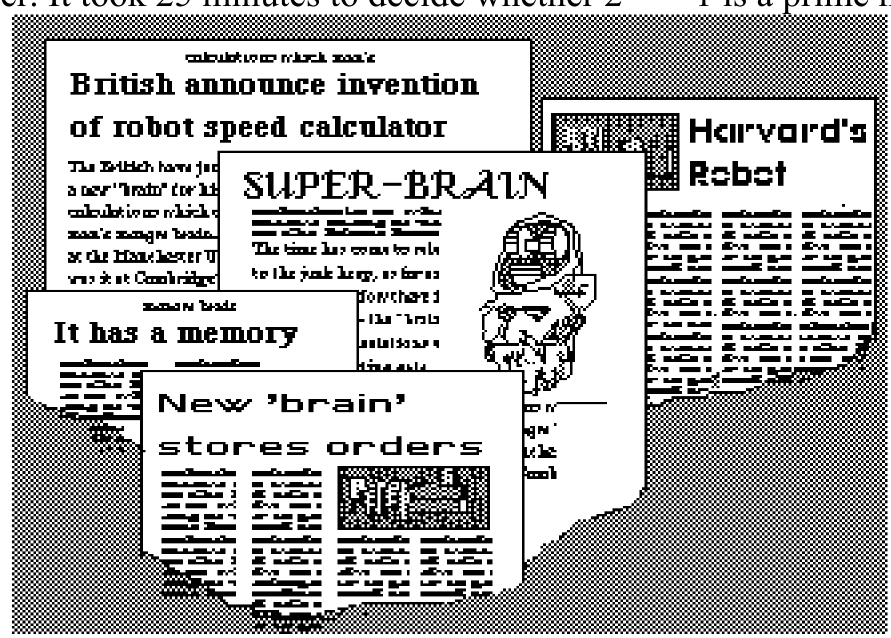
«*Eckert-Mauchly were not too happy! Who really invented the computer? Who really had patent rights? E-K applied for a patent in 1947. Battles ensued for years, between Neumann, Eckert-Mauchly, Moore School and, later, patent infringement battles between IBM and CDC, and Honeywell and Sperry Rand, etc. The ENIAC patent was finally issued in 1964. In 1972, Judge Earl Larson (no relation to the current chair of CS!), being of sound mind and in possession of some 40,000 documents (including von Neumann's infamous Draft Report) ruled that the computer was invented by John Vincent Atanasoff! (... but that's another story).*»

However, it would be years before the first stored program Eckert-Mauchly computer would be available. Desperately short of funds, their corporation was absorbed by Sperry Rand Inc. and, based upon their EDVAC design, the Universal Automatic Computer (UNIVAC) was eventually built.

«*The first production model was used by the U.S. government for the 1950 census. Grace Hopper was the premier programmer for Eckert-Mauchly, and subsequently the Univac division of Rand.*»

******

In July, 1948, the Manchester University Computer (MUC) was unveiled. The innovative memory, using cathode ray tubes, was designed by a British radar Engineer, Freddie Williams, freed from the activities of the war. Although small, it was the first fully electronic, *stored program* computer. It took 25 minutes to decide whether $2^{127} - 1$ is a prime number.[d]



The newspapers were delighted, yet there was rampant skepticism as to what a no-purpose machine could do. A machine designed to calculate

---

[d] a problem clearly designed to impress the general public (!)

shell trajectories? Okay. A machine to compute star maps? Okay. But a machine without purpose?

Inspired by the EDVAC design (and having attended lectures at the Moore School, obtaining a copy of von Neumann's *Draft Report*), a young Cambridge Physicist, Maurice Wilkes directed the building of the "user-friendly" Electronic Delay Storage Automatic Computer, the EDSAC.
«*The "Delay Storage" referred to an electromechanical delay line used for storage: oscillating quartz crystals generating pulses in tubes of mercury, the pulses recycled to provide memory. (In place of mercury, Turing had suggested gin and tonic because the speed of propagation was relatively insensitive to temperature changes.)* »

The EDSAC was a fully electronic general purpose, *stored program* computer, in operation by 1949 (before EDVAC, upon which the design was based). It contained some 3000 vacuum tubes, could perform additions in 1.4 milliseconds and emphasized "programming". Indeed, EDSAC could access a library of programs called (would-you-believe) *subroutines* including what was thought impossible at the time: a subroutine for numerical integration which (by calling an "auxiliary" subroutine) could be written without knowledge of the function to be integrated! A problem: whenever a tape was read, the subroutine might not go to the same memory locations, so certain addresses had to be changed.[e] This was overcome by preceding each piece of code with a set of "co-ordinating orders" (making it self-relocatable). Later, the Cambridge team devised symbolic addresses whereby the programmer used some label and the computer assigned the memory addresses when the program was read (avoiding the need to change any explicit addresses which might appear in the code).

But Wilkes found few who appreciated the problems in programming ... until he visited the U.S. in 1950 and met Grace Hopper's group. (On the way he stopped off at the University of Toronto to give a talk, at Kelly Gotlieb's invitation). During his U.S. trip he also argued with Howard Aiken (of Harvard-IBM *ASCC* fame) of the advantages of a *binary* computer (as opposed to a *decimal* machine).

Although EDSAC solved a variety of problems (including a nonlinear differential equation proposed "in characteristically barbed fashion" by the distinguished statistician R.A. Fisher ... much to Fisher's surprise), in order to avoid the suspicions associated with *analog* machines, EDSAC spent some time computing primes (which no analog device *could* do). It

---

[e] Each instruction involved an operation to be performed and a single memory address identifying the location of the operand. Later, multiple address instructions would become popular.

impressed at least one sceptic: number theorist L.J. Mordell (who later obtained an honorary degree at the University of Waterloo). In 1951, Wilkes co-authored the first book on computer programming. Addison-Wesley, feeling such a book would be risky, offered no royalties until 1000 copies were sold; they were, by the following year.

******

By 1950, MIT had nearly finished the *Whirlwind* computer[f] which featured (for the first time): graphical output terminal, light pen interaction between operator and machine, data communications over telephone lines and later, a magnetic core memory.

«*This made possible, for the first time, very large memories (in spite of von Neumann's claim, in 1955, that a memory over 10,000 words for the Univac LARC was a "waste of the government's money").*»

In 1954 IBM had completed the Naval Ordnance Research Computer (NORC) which performed additions in 15 microseconds, multiplications in 31 microseconds. It handled numbers in the form of *words* consisting of 16 decimal digits:13 for the number, 1 for the sign, 2 for the exponent. It had eight magnetic tape units with 1200 feet of tape at 510 characters per inch, feeding information to the central computer at 70,000 characters per second. Words were available, from memory, in 8 microseconds.

W.J. Eckert, head of IBM's Watson Scientific Computing Laboratory, had supervised the construction of NORC. He said: "... we expect a billion operations between errors ...".

«*Thomas John Watson (1874-1956) was a business executive who, in 1914, became president of the Computing-Tabulating-Recording Company (which grew out of Hollerith's "Tabulating Machine" company). In 1924, the company was renamed: International Business Machines. In 1953 the Univac was practically the only computer available to a commercial firm. Within two years IBM would sell more than half the computers in the US. Within another two years the computer industry would be known as "IBM and the Seven Dwarfs". Sperry Univac was the first dwarf. Some Sperry Univac people jumped ship and created the second dwarf: Contol Data Corporation. One who jumped ship was Seymour Cray, who build the CDC 1604 - the world's most advanced large computer.*[g] *Cray eventually left CDC to build his own computers.* »

Computers could now perform complex calculations with *speed* and

---

[f] MIT was taken by surprise by U of Pennsylvania's ENIAC and decided to switch from building an analog machine to a digital computer.

[g] The CDC people who left Univac had designed the Univac 1103, CDC's address was 501 Park Ave., and 1103 + 501 = 1604.

reliability.

******

One of the most forward thinkers of the day thought scientists had missed the point. To use computers for arithmetic was a waste.

A. M. Turing had published a paper in 1936: *On computable numbers with an application to the entscheidüngsproblem.* In it he defined a theoretical "computing machine" (to become known as a Turing Machine) which, in principle, could compete with a human in performing cerebral (logical) tasks.

During the war, Turing put these ideas to work; he was engaged in breaking the German codes at the highly secret Bletchley Park.

«*Bill Tutte (distinguished professor, C & O dept. at UW) also worked at Bletchley Park during WW II.*»

In particular, Turing was put in charge of breaking the ENIGMA which encoded messages (concerning daily German troop movements) not by a fixed transformation from message to code, but by *continuously* altering the characters using rotating toothed wheels. Although Turing worked out a strategy for deciphering the messages, there was too little time.

Alan M. Turing

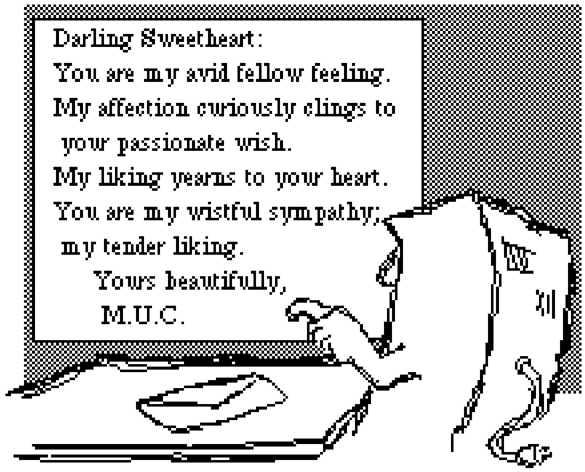
So an electronic (vacuum tube) *special purpose computer*, the COLOSSUS, was built according to Turing's design. It began operation in 1943 when work on ENIAC had just begun. It was highly successful (some believing that "Colossus won the war"). Secrecy prevented any knowledge of its capabilities for some thirty years.[h]

Turing was convinced that computers could do far more than carry out a sequence of instructions, far more than perform mere arithmetical calculations: computers could learn. After the war he built the *ACE* computer to simulate human thought. The ACE "pilot" model[i] was completed in 1950 and regarded by many as the world's most advanced computer.

---

[h] The 1960 Colliers encyclopedia identifies the 1944 Harvard-IBM Mark I as the *first general purpose automatic digital computer.*

[i] it has been said that the Bendix G-15 was very much like the pilot ACE.



Alas, it was used solely for scientific number crunching. Turing was disgusted with this, and with the bureaucracy, and joined the University of Manchester where he designed a machine to work in conjunction with the Manchester University Computer to do "creative word processing", including love letters.

In 1950, Turing proposed a test of intelligence: for five minutes you ask questions of an unknown "device", either a machine or a human (presumably intelligent). If you are not 70% sure that the device is a machine then you must concede that the device has *some* intelligence. Turing predicted that machines would pass such a test of intelligence by the year 2000.

By 1957 (when our *Computer Science at Waterloo* begins) there were dozens of suppliers of computing devices, all designed to perform rapid and accurate calculations. In addition to these "automatic digital computers" there were a variety of *other* machines, *not* digital computers, which could handle one of: bookkeeping, data sampling, radar fire control, file searching, flight simulating, machine tooling, telephone switching, navigation systems, music generation, game playing (tic-tac-toe, checkers).

But Turing never lived to see the myriad uses to which today's *fully electronic, automatic, general-purpose, stored-program computers*[j] are put ... including *all* of the above.

In 1954, prosecuted for homosexuality, Alan Mathison Turing committed suicide.

---

[j] I stress the nature of the beast since arguments concerning "who invented what" (and they continue unabated to this day) might be circumvented by just such elaborate prefixes ... as well as a clear definition of the word "computer"! (PJP)

Chapter One

the First Decade 1957-1967

It was 4:27 p.m. The meeting of the board of governors had but one item on the agenda: the recommendation of the Academic Advisory Committee. Ralph Stanton leaned back, gazed at the clock, then at Ted Batke; both looked weary. Carl Pollock was speaking in favour of the recommendation, but there seemed little hope of other support from the board.

******

Little more than a year had passed since Ralph left the University of Toronto to chair the mathematics department at Waterloo College, a small liberal arts college sponsored by the Lutheran Church and granting degrees through an affiliation with the chartered University of Western Ontario. For members of the math department at the College it meant the curious, annual ritual of driving to London, Ontario, to have final examination papers approved by the math Chairman at Western.

But there was something else: co-operative education, a first in Canada. It was a novel scheme championed by Joseph Gerald "Gerry" Hagey, late of the B. F. Goodrich Company and current President of Waterloo College, and Ira G. Needles, president of B. F. Goodrich and chairman of the board of governors.

Classes had begun in July, 1957, with some seventy-five students who had completed Ontario grade 12 and enrolled in a "pre-engineering" program on the small campus, alternating 3 month academic terms with 3 month work terms. They constituted the *Associate Faculties* of Waterloo College. The church-affiliated College now had a non-denominational component (a Faculty of Applied Science, with Les Emery as Principal) which could attract provincial funding not available to sectarian institutions.

In 1957, classrooms for the *Associate Faculties* were located in two prefab huts, Annex 1 and Annex 2, providing some 8,000 sq. ft. of teaching space behind Willison Hall, the only permanent structure on campus besides the Arts building. Ralph Stanton had hired four graduates from the Engineering Physics program at the University of Toronto: Dave Fisher, Jim

Leslie (now in Physics at UW), Keith Oddson and Peter Ponzo (now in Applied Math at UW)[a]. In 1958, A. S. "Bert" Barber succeeded George Dufault as director of coordination, handling job procurement and sharing space with the fledgling math department in a house at the Albert and Dearborn corner of campus.

******

But, in the fall of 1957, the recommendation of the Academic Advisory Committee (AAC) constituted a necessary preamble to the creation of a new university - and support was minimal.

Ralph Stanton brushed the hair from his forehead, leaned on the table with chin in hands and listened to the debate. It was 4:42 and the meeting was to end at 5:00. p.m. A.M. Snyder, a local businessman and member of the board, had spoken vehemently against the recommendation. Now Carl Pollock was speaking, again. A native of Kitchener and president of Electrohome Canada Ltd., his support was needed, and appreciated, yet his seemed the only supportive voice. Gerald Hagey had been strangely quiet. Ira Needles, as chairman of the Board, had spoken once or twice; it was clear that support did not lie with him.

Ralph closed his eyes. It was nearly a month since he and Ted Batke and the other members of the "fearsome five" AAC (Ted Batke, Ron Bowman, Art Cowan, Bruce Kelly and Ralph Stanton) had tramped the few blocks down Dearborn Street to the Schweitzer farm, past the vacant lots to the fields of corn and mud. The chestnut trees hung solemn above the farm house and dark clouds billowed in a grey sky. It was cold and bleak, yet there was a certain measure of excitement: the property was for sale, the price was reasonable, it was less than a country mile from the Waterloo College campus.

It was clearly impossible to develop a university on the small campus of a sectarian College. The Schweitzer farm could become, *would* become, the home of a new university. It meant applying for additional funds from the province and, in particular, a charter to grant degrees. Yet, weren't there already students studying in the new co-operative education environment? Didn't the College already have a complement of arts students? Surely the provincial government would support the application for a charter - and

---

[a] The "College" was *small*. We four threw a house party and invited all faculty, staff and students (the students making garbage cans full of potato salad). Hagey walked in ... and left soon after. (PJP)

provide the necessary funding. The timing was right. *Every qualified youth must have access to a university education*. It was becoming a familiar cry in the province.

It had taken less than 24 hours for the AAC to agree on the wording of the recommendation: purchase the Schweitzer farm and expand the facilities of the College to accommodate the increased demand for higher education in Ontario. They had approached Gerald Hagey and he had agreed to put it before the Board of Governors of the *Associate Faculties*.

******

Yet, it now seemed hopeless.

At 4:55 there was a rustling of papers in the board room. The meeting was coming to an end. The recommendation would be ignored; too expensive, too grandiose. The AAC had been promised an hour, and the hour had ended. There was little more to debate. Members of the Board were collecting their notes.

Then A.R. Kaufman said something, quietly, unobtrusively ... and the rustling of papers stopped.

Kaufman was undoubtedly the most respected member of the board. A wealthy industrialist, philanthropist, president of Kaufman Shoes, outspoken proponent and supporter of family planning (long before it became fashionable), he commanded the admiration and, indeed, awe of the community and, more importantly, the board room.

Gerald Hagey looked at the tall, gaunt man who had sat for an hour without saying a word.

All eyes turned to Kaufman.

There was silence in the room.

"Gentlemen," Kaufman said softly, "I think we should buy the property ... and we should do it soon."

The silence continued for several seconds, then the chairman harumphed and laughed and chuckled - and agreed with Kaufman. Then all agreed.

The debate had ended.

The Schweitzer farm would be purchased and a new university would be created.

******

- By the end of 1958, Dr. T. L. Batke, Chairman of Chemical Engineering, saw the first building rise from the mud: chemistry and chemical engineering (at a cost of $780,200).

- The mathematics department moved into the Schweitzer farm house and the prefab classrooms were moved to the new site.
- Within a year the Physics and Mathematics building was completed.

- Finally, in 1959, the provincial government blessed the buildings and stucco farmhouse and 237 acres.
- It was incorporated as the *University of Waterloo*.

The Schweitzer house, with the addition (rear right), is now the Grad House.

- Waterloo College and the Lutheran Synod of Canada had debated the issue ... and decided against affiliating with the new institution. Indeed, the student newspaper of the College openly questioned the "validity and prestige" of a degree from a university other than the long-established University of Western Ontario.

- Gerald Hagey "left" the College to become the first president of the University of Waterloo.

- Ted Batke became academic vice-president.

- Ira Needles and Carl Pollock become charter members of the board of governors of the new university.

- In 1959, the Honorable Dana H. Porter, Chief Justice of Ontario, was installed as the University's first Chancellor.

- Douglas T. Wright became Dean of Engineering (and, much later, President of the University of Waterloo) and Bruce Kelly[b] became Dean of Science, later succeeded by W. A. E. "Pete" McBryde.

---

[b] Bruce Kelly died in office.

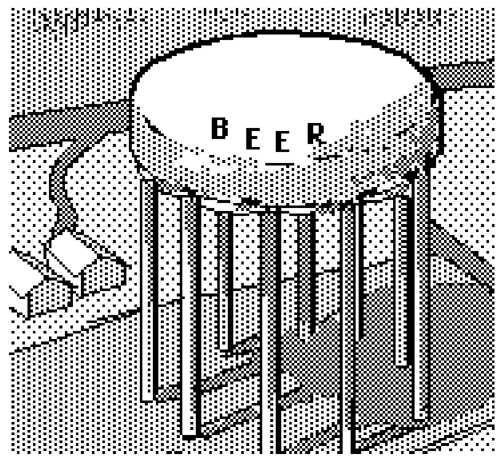
- In 1960, a Faculty of Arts was established, the earlier notion that Waterloo College would play that role having been dismissed ... and Mathematics became a department within that Faculty.

- The 3 month co-operative terms were changed to 4 months so that two terms more nearly matched the academic year.

- By 1962, the university had the second largest Engineering enrolment in Ontario (846 undergraduates) and the third largest in Canada.

- Dearborn Street, the dirt road intersecting Albert, was eventually paved and renamed University Avenue.

One day in June of 1958, in the early hours of the morning, a group of Engineering students marched to Lester and Sunview Streets. Residents awoke to find a four-letter word on the water tower.



******

Building proceeded at an astonishing pace (exceeding any other university in the country), co-operative education attracted large numbers of high school graduates, the grey fields grew verdant, mature trees were planted, plans to introduce freely-roaming peacocks were considered, the campus ring road held an expansive promise of things to come ... and an additional 733 acres were purchased across Columbia Street. There were tentative thoughts that this "North Campus" might accommodate a second, associated university with year 1 common to both campuses, perhaps relegating all co-operative programs to this "other" campus.

«*The "peacock" idea was championed by E. Mike Brookes (superintendent of buildings and grounds). Brookes appears later in this CS History (CSH), 20 years later, in connection with the Oxford English Dictionary project*.»

Waterloo Lutheran College had been granted a charter and became Waterloo Lutheran University (WLU), cutting its ties to Western. Within a few years the Lutheran Church accepted government funding and WLU was renamed[c] Wilfred Laurier University.

Within a year of the enrolment of the first engineering students in the *Associate Faculties* of Waterloo College, Ralph Stanton had begun a graduate program in mathematics. In 1960, the University of Waterloo held its first formal convocation, and the first degrees granted were *all graduate degrees in mathematics*. Ronald C. Mullin received the first earned degree: an MA in mathematics. (He later obtained a PhD in mathematics at Waterloo, and was later to become Chairman of the department of Combinatorics & Optimization.). MSc degrees in Applied Mathematics were awarded to J. Douglas Lawson (who later became Chairman of the Computer Science department) and to Peter H. Roe, (later to become Associate Dean of Engineering).
«*Mullin received the first UW degree because MA's were awarded before MSc's!* »

In 1960, Ralph Stanton became Dean of the Faculty of Graduate Studies and extended programs in mathematics to the doctorate level. By the end of the first decade, the department of Mathematics was to have 89 graduate students and well over a thousand undergraduates.

By 1960/61, the Arts Faculty had twenty-four faculty members; *half were in mathematics*, including C. F. Arthur Beaumont who later became the first Associate Dean of the Math Faculty (undergraduate studies), Kenneth D. Fryer who also became an Associate Dean (external affairs), Douglas G. Wertheim who later became the first Chairman of Pure Math (all three joined the department from the Royal Military College), David A. Sprott from the University of Toronto (who later became the first Dean of Mathematics) and J. Wesley Graham of IBM (who became the first Director of the Computing Centre), the latter two having been former students of Ralph Stanton. The Mathematics department was the first on campus to initiate visiting professorships and hosted an impressive array of distinguished scholars, including Janos Aczel who later joined the department (and became, with W.T. Tutte, a *distinguished professor*).

Ralph Stanton introduced a course in numerical analysis, Math 132; he taught the course and wrote the textbook. It involved extensive computational methods with mechanical calculators and was the beginning of an abiding interest in computation at Waterloo.

---

[c] It was rumoured that the name was chosen to avoid replacing the large WLU sign on the campus lawn.

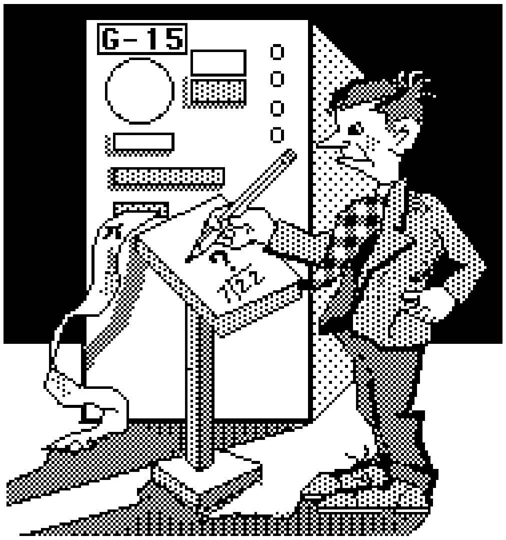
******

J. Wesley Graham placed the phone on the hook, slowly, absentmindedly, leaned back and stared out the window at the Toronto skyline. He had joined IBM shortly after graduating from the University of Toronto and was looking forward to a promising future. It was a dynamic and vigorous company and he was now manager of the Applied Science Division with some hundred staff members. Yet, that phone call could change everything: he had been asked to join the math department at the newly established University of Waterloo, and he had been asked by the Chairman, Ralph Stanton.

He remembered Ralph as a professor of mathematics at the University of Toronto; a great instructor, a real character. But leave IBM and get away from computing, entirely? The more he thought of it, the more he remembered the headaches and endless trips around the country. Perhaps it wasn't such an absurd idea. A math prof. Not a bad idea at all. Leave computing behind, and teach math? Why not?

In the Fall of 1959, Wes left IBM to become a math professor. His pleasure was not shared by all in his family. He brought his mother-in-law to the newly acquired acres of dirt at the end of Dearborn Street and she stood on a hill and cried.

But Wes didn't get away from computing - not entirely. Ralph Stanton immediately had him teaching Math 132 (computing/numerical analysis), put him on the university computer committee (chaired by Basil Myers, Chairman of Electrical Engineering) and asked him to arrange for delivery of a digital computer. It arrived on January 4, 1960, and stayed for one week: a Bendix G-15, on loan, for evaluation.



The Bendix sales rep was present during that week and Ralph came to him each day to pose a math problem. The rep would laboriously program the G-15, and Ralph would take out his pad and pencil and check the answer.

In January, 1960, the UW computer committee prepared a document entitled COMPUTER NEEDS OF THE UNIVERSITY, in which it recommended that a *small* digital computer be ordered, immediately, to support teaching and research. To this report was attached a list of machines currently in educational institutions across the country (see TABLE I, below).

But still the big question: *which machine?* Later that month, Wes prepared a technical report comparing the G-15 to the "newest and most modern machine available", the IBM 1620, in which he states:

- *The G-15 uses a rotating magnetic drum to store approximately 15,232 decimal digits of data and instructions.*
- *The 1620 memory is constructed from magnetic cores and stores 20,000 decimal digits of data and instructions.*
- *The G-15 employs high speed loops, called "short lines", to help overcome some of the drum access problems. In a large number of cases this feature considerably improves the timing but far from eliminates the access problem.*
- *The G-15 utilizes a one-digit binary serial adder to complete all of its arithmetic instructions.*
- *The 1620 utilizes tables stored in memory*[d] *to complete the addition, subtraction and multiplication commands. Division may be programmed.*
- *The G-15 input tape is searchable and contains 2,500 words of data. It operates at a speed of 250 characters per second. Output is at 17 characters per second on punched tape.*
- *The 1620 input tape is read at 150 characters per second but is not searchable in the same sense as the G-15.*
- *Neither the G-15 nor the 1620 have floating-point hardware. The respective*

[d] I read somewhere that the 1620 was known as the CADET: Can't Add - Doesn't Even Try. (PJP)

manufacturers quote the following operating times:

	1620 (8 digits)	*G-15 5 dig.*	*G-15 12 dig.*
Add	*7.5*	*29.0*	*58.0*
Subtract	*7.5*	*29.0*	*58.0*
Multiply	*15.0*	*29.0*	*58.0*
Divide	*61.0*	*29.0*	*58.0*

All times quoted in milliseconds.
All other instructions (except input-output) are relatively fast (usually less than one millisecond).

- *The 1620, being a core machine, has all data immediately available to the arithmetic unit at all times.*
- *The G-15, however, on the average must wait for one-half of a drum revolution for data access.*
- *The G-15 has the following programming systems:*
  INTERCOM: - an interpretive coding system
  POGO: - compiler
  ALGO: - algebraic compiler

- *The 1620 has:*
  SPS: - Symbolic Programming System
  FORTRAN: - Translator

In February, 1960, Basil Myers sent a memo to president J. G. Hagey. It recommended that an IBM 1620 be leased, to support teaching and research. Leasing would cost $640/month, including servicing and maintenance; this was 60% less than the normal rental fee. Outright purchase would cost $59,600 including a 20% educational discount.

TABLE I

COMPUTER FACILITIES IN EDUCATIONAL INSTITUTIONS
DECEMBER 1959

NAME OF INSTITUTION	COMPUTER FACILITIES OWNED OR RENTED BY THE INSTITUTION: SMALL DIGITAL	LARGE DIGITAL	ANALOG
University of Alberta	1	1*	2
Assumption University	1*		
Univ. of British Columbia		1	1
McGill University		1	1
McMaster University	1		
University of Manitoba	1		2
University of New Brunswick	1		1
St. Francis Xavier University			1
St. Mary's University	1		
University of Saskatchewan	1		1
University of Toronto	1*	2**	1

University of Western Ontario		1	1
Ecole Polytechnic	1		
Ontario Agricultural College	1		
Royal Military College			1
University of Ottawa		1	2
Nova Scotia Tech			1
	10	7	15

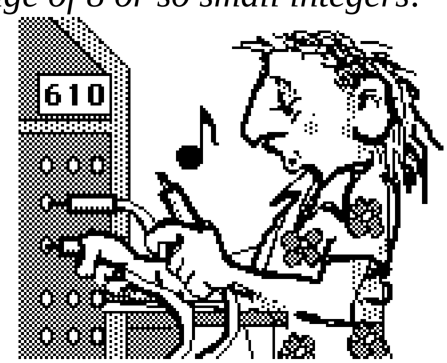
* denotes machine on order
** denotes 1 of the 2 machines on order

Delivery time for the 1620 was to be 14 months; the computer committee was impatient. Wes knew that IBM had a 610 in *inventory* and arranged to have this machine installed on an interim basis (for a rental fee of some $200/month, sufficient to cover the cost of maintenance). Thus, in early 1960, an IBM 610 arrived.

The University of Waterloo had its first digital computer.

Barry Wills (currently a professor in the Systems Design department[e]) was a student when the 610 arrived. He recalls:

"UGH! is the four-character word which best describes my experiences with the 610 in my undergraduate EE labs. Although we were perhaps the first engineering undergraduates in Canada to have a computer lab, it was painful. The 610 had a patch panel for its programming and a paper tape for some input. I remember spending hours and hours to get a programme going which would find the average of 8 or so small integers! Hardly a breakthrough! Nevertheless, many of my classmates (stone-age hackers) were very taken with the 610. Fortunately the 1620 arrived. It was, relatively speaking, a "gem", in spite of a paper tape reader which frequently ripped the sprocket holes out of your input tape, rendering it (an appropriate word) useless. Later we got a keypunch and a wild Rube Goldberg device called a card-to-tape converter. However, there was still a high probability that if the converter didn't eat your cards the tape reader would mutilate your tape!"



Peter Roe recalls Doug Lawson spending hours programming the

[e] Although the first degrees at UW were graduate math degrees, eventually bachelor's degrees *were* awarded! Barry was among that first graduating class, in Engineering.

machine to evaluate a cubic polynomial.

Bev Marshman (now a professor in Applied Math) notes that, in the early days, the faculty was very protective of the 1620:
"Wes Graham caught me in the computer room, about to insert a deck of punched cards. He chewed me out vociferously - students were forbidden to run their own programs. Later, Wes recommended me for my first off-campus job, at the University of Guelph."

Later, a sign would be hung on the machine (reflecting the predominantly German makeup of Kitchener-Waterloo):

Achtung!!
Alles lookenspeepers. Das computenmachine is nicht fur gefingerpoken und mittengrabben. Ist easy schnappen der springenwerk, blowenfusen und poppencorken mit spittzensparken. Ist nicht fur gewerken by das dummkopfen. Das rubbernecken sightseeren keepen hands in das pockets - relaxen und watchen das blinkenlights.

******

The Jan. 1960, report of the computer committee admitted that the need for a sizeable *analog* computer installation was not urgent. Departmental units (such as provided by Heathkit) would suffice for the current undergraduate load.

However, the report did emphasize that a *large* digital computer would be required within three years. The problem was money: the highly successful IBM 650 would cost 1 to 2 million dollars, the Honeywell 800 was selling for 2 1/2 million and the Univac Larc for 6 million. (While IBM was still marketing the vacuum-tube 650, Univac had a transistorized Larc.) Four universities in Canada already had 650's, the University of Alberta had ordered an IBM 704 and the University of Toronto, in addition to its 650, was building an Illiac II (designed and built at the University of Illinois). In order to avoid "forever playing second fiddle to Toronto", the computer committee seriously considered building a modified Illiac I. Michigan State was doing precisely this (the chief modification being the replacement of a drum memory by a magnetic core). The cost would be low (an estimated $200,000), it would provide "many graduate thesis and ancillary research projects" (particularly important since the university was "in the process of establishing a Faculty of Graduate Studies", with Ralph Stanton as Dean), and it would have the further advantage that "we could build an extremely modern machine - better than many of the existing commercial designs".

Although room 207 in the Physics and Mathematics building was considered (to house the machine), it was anticipated that "we shall eventually need a separate computer building".

The hiring of a design team, beginning in the summer of 1960, never materialized - and a modified Illiac I was never built.[f]

******

Don Cowan pulled the car to the curb at the corner of King and Erb and rolled down the window.

"Can you tell me where to find the University of Waterloo?"

The pedestrian frowned. "The what? The University of who?"

Don looked at Marg, shrugged, and asked: "How about Waterloo College?"

The pedestrian smiled eagerly. "Sure ... just down to Dearborn and turn left."

Don had graduated recently with a B.A.Sc. from the University of Toronto and, along with several of his classmates in Engineering Physics. While in first year (1956) he had listened to talks by U of T faculty indicating bright futures in *their* area. They all promised great things: "Our labs are the best in the country." "We're doing fundamental research in geology." Only Ralph Stanton had promised nothing more exciting than a supply of paper and pencils. The following year Stanton would head for Waterloo College, and the idea of teaching at a new university seemed exciting[g] - so in 1960 Don headed off to Waterloo, with his wife, to enrol in the graduate program and start lecturing in the "computer division" of the department of mathematics. Alma Fielding, the departmental secretary, introduced Don to Wes Graham: "Wesley *dear*, would you mind showing this gentleman around?"

While lecturing, Don took graduate courses in mathematics, considered joining Electrical Engineering ("I bugged Basil Myers for a lectureship in EE"), then was asked by Ralph Stanton to *look after the computers* (for an incremental increase in salary).

---

[f] In 1959 I took a graduate course from Kelly Gotlieb at U of Toronto. Although it was called "logical design of digital computers", it *could* have been called "design of the Illiac". I can recall, later, programming the Illiac I at U of Illinois. It completely filled a small room, hot from the glow of vacuum tubes. (PJP)

[g] His mother-in-law wasn't so excited. She had read about a university in Waterloo and couldn't believe that her daughter was being hauled off to WLU.

"I taught courses to the Engineers because Ralph wouldn't trust me with his baby, Math 132," says Don. *"Then I backed into computer science. Wes challenged me to write a one-card program which would read an instruction into the 1620 core memory. Ralph would give his students a problem in numerical analysis which required a number of iterations. We would punch up the code for a single iteration on a short piece of paper tape, glue the ends of the tape together, then run this loop through the reader, continuously, until successive iterations agreed - at least enough to satisfy Ralph."*

Don was hooked on computing. When it was learned that the 1620 memory was to be increased from 20K to 60K, Don was asked to modify the compiler to handle the extra memory. As was usual at Waterloo, the modification was to be done before the extra memory arrived, so Don drove to IBM (New York) to test the compiler, with the modifications on a deck of cards. It worked. When he returned to Waterloo, it didn't work.

The headaches had begun.

- Gus German (a math undergraduate) found his way into the locked 1620 room and fouled the paper tape reader by unspooling the coil of tape from the inside.
- The machine "radiated", so students would write code which played "Yankee Doodle Dandee" on any nearby radio.
- The card reader sucked in the elastic band around a card deck - the day before the computer was to be used for a visiting group. A car was prepared to run back and forth between Waterloo and the University of Guelph (which had a working reader), but the reader was fixed in time.
- In spite of extensive visits by Ralph Stanton and Ken Fryer (to promote studies in computer science), high school teachers were asking, in greater numbers, to visit Waterloo and learn about computing. Wes approached Ralph with the idea of "Computer Science Days" which would formalize these visits. Ralph said it was "the stupidest idea yet", then, the same day, suggested it himself (describing his idea as *brilliant*). It would be a day-long affair ... and Don ran the first hectic Saturday, giving all the lectures, overseeing the programming labs and answering questions about the 1620. He had become a computing guru on campus.

Don eventually enrolled in the PhD program and, in 1965, obtained his degree in mathematics under Gerald Berman (who later became the first Chairman of Combinatorics & Optimization).

******

In 1962, Ted Batke (as Vice President of UW) appointed Wes Graham Director of the Computing Centre which had now become a separate department in the university. (Earlier, it was an appendage of the math department, with Basil Myers as Director and Stanton as Associate Director.) Ted explained that Wes was to maintain normal professorial duties; however, there *was* a $50/month stipend associated with the job.

Ralph Stanton became Associate Director (... no stipend associated with this post!).

Shortly after the 1620 arrived, UW obtained FORGO from the University of Wisconsin. Wes Graham notes that FORGO was the proper forerunner of WATFOR and recalls:

"Sometime in 1962, Ralph Stanton visited the University of Wisconsin ... maybe to browse through their old book collection ... Ralph was a nut about old books[h]*. Anyway, a math student, Gus German, went along and discovered that Wisconsin had written a FORTRAN processor called FORGO. Gus brought it back on a deck of cards and we installed it on the 1620. A program which took an hour to compile could now be processed in under a minute. It was 100 times faster. I was really impressed."* (Don Cowan is convinced that Gus "pirated" the deck of cards. Later, UW acquired an approved version.)

Bob Zarnke (a student, at the time) recalls: *"The compiler provided by IBM* (for the 1620) *consisted of two card decks. You loaded pass I, with your source program behind it. After reading the source program, it punched an intermediate deck of cards (in binary). You then loaded pass II with the intermediate deck behind it. Pass II then punched an object deck, which you were finally able to run. Any output produced by the program was punched and subsequently listed on the 407. With FORGO, you just loaded your source program behind the compiler and it compiled and executed without intermediate decks."*

Students would punch up their own cards, wait in line to read in their programs and have them run in about 15 seconds. The 1620 was available

---

[h] I visited Ralph in Wisconsin. He was there to give a lecture series. (PJP)

from noon to 2 p.m. weekdays and could now execute some 200 jobs/hour.

Cafeteria style service was to become the popular style for years to come.

******

By Spring, 1963, the 1620 was purchased outright (and available around the clock to faculty and students) and an IBM 1710 (a 1620 with D-to-A and A-to-D converters) was ordered. But the demand for computing time increased, the rooms became crowded (and there was thought of filling the halls of the Physics and Math building with card readers). Nearly every student on campus took a course in computer science. It was clear that a much larger machine was needed and Graham recommended an IBM 7040 and an IBM 1401 ("which had a nice printer attached to it").

In June, 1964, the UW Quarterly Report heralded the imminent arrival of a "new, high-speed, research computer valued at more than $1 million". The IBM 7040 would arrive in the Fall (and was "approximately 100 times as fast as the 1620 model currently in use") along with a medium-sized IBM 1401 which would handle input-output. Peripherals included a card reader, eight tape drives, a disk file storage system and a high speed printer. The 7040-1401 system was the first such integrated installation in Canada and, by December 1964, the Computing Centre (located in a five-room corner of the Physics and Math building) was one of the most sophisticated in North America. The total cost of $2 1/4 million was more than the cost of the building itself!

When the 7040 arrived (Fall, 1964), president Hagey read about it in the *K-W Record* and phoned Wes Graham: *"Wes, I'm amazed. Basil always told me we needed a 2 million dollar computer - and now we have one. I didn't even know we ordered it. I sure like the way you do things."* [i]

******

Another CS story begins around 200 B.C., in Sicily ...

He stood on a rock, his robe billowing in the light wind that ran from the sea, and he spoke to the mathematicians of Alexandria who were gathered in Syracuse. The figures in the sand were erased by the wind, yet he continued.

---

[i] Ralph Stanton doesn't believe this story!

Compute, O friends, the host of oxen of the sun - if thou has a share of wisdom. Compute the number which once grazed upon the plains of Sicily and which were divided according to colour into four herds: one white, one black, one spotted and one yellow. The bulls formed the majority of animals in the herd and the relations between them were as follows:

Archimedes stood down from the rock and wrote in the shifting sands:

$$W = \left(\frac{1}{2} + \frac{1}{3}\right)X+Z, \quad X = \left(\frac{1}{4} + \frac{1}{5}\right)Y+Z, \quad Y = \left(\frac{1}{6} + \frac{1}{7}\right)W+z, \quad w = \left(\frac{1}{3} + \frac{1}{4}\right)(X+x),$$

$$x = \left(\frac{1}{4} + \frac{1}{5}\right)(Y+y), \quad y = \left(\frac{1}{5} + \frac{1}{6}\right)(Z+z), \quad z = \left(\frac{1}{6} + \frac{1}{7}\right)(W+w).$$

If thou canst give, O friends, the number of bulls and cows in each herd, thou art not unknowing nor unskilled in numbers, but still not yet to be counted among the wise. Consider, however, the following additional relations between the bulls of the sun.

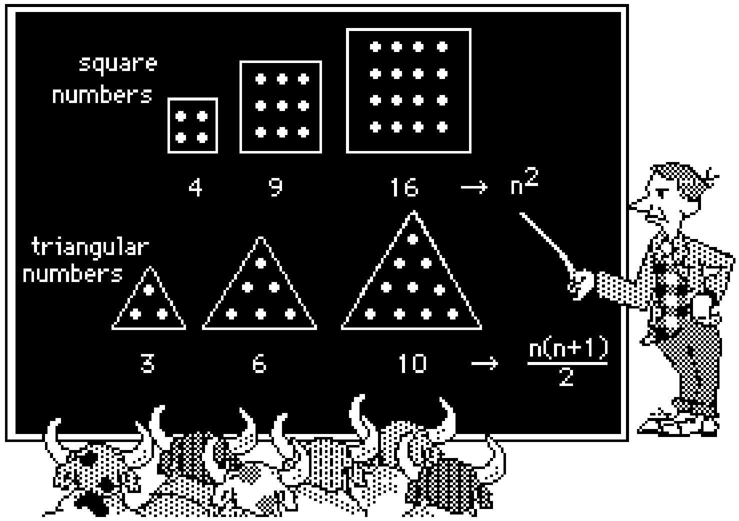
And he wrote again in the sand:

$W+X$ = a square number, $\quad$ $Y+Z$ = a triangular number.

When thou hast computed the totals of the herds, O friends, go forth as conqueror and know that thou art most skilled in the science of numbers.

******

It was Spring, 1965, and Ralph Stanton stood before three third-year math students, chalk poised, pink tie flailing a worn jacket. He described the *cattle problem of Archimedes* and the students listened intently. Various attempts had been made in the past, but, for more than 2000 years, no "complete solution" to the problem had been found.



Now, Angus German, Robert Zarnke and Hugh Williams were to attempt a solution.

The new IBM 7040 had been on campus for almost 6 months and would be the ideal tool to tackle the problem; it was capable of 62,500 additions of 11 digit numbers per second.

The 3-man team cogitated for a month on the method of solution (complicated by the fact that the 7040 had only 32K words of core memory), and, with Wes Graham's permission/encouragement, used the computer *after hours*. By June, 1965, they had the answer. The 7040 had worked for hours; the computation was completed on the 1620. Total computing time: seven hours and 49 minutes (most of it to convert to decimal for printing!). However, the margin of this *history* is too small to hold their result, namely the sum of the eight integers W, X, Y, Z (the number of bulls) and w, x, y and z (the number of cows). The sum is 206,545 digits long and, if printed 10 digits/inch, would extend for one-third of a mile.

The student, Hugh Williams (who went on to become a computational number theorist at the University of Manitoba), noted (in 1965): *the answer is so large that if the universe were filled with the smallest particles of matter, the number of such particles still wouldn't come near the answer.*

Various newspapers carried the story:

"Cattle problem solved after 2200 years."
"2000-Year-Old Math Problem Is Solved."

A description of the solution (and the high speed computing techniques devised to generate the answer with memory limitations) were published in *Mathematics of Computation* (with a statistical analysis of the digits, but without the number itself!)

******

Alas, the 7040 was less useful than the 1620 (which, at least, had the Fortran processor FORGO); the 7040 took a minimum of 2 minutes to run each job. (There was insufficient core memory to hold the compiler. It was read in piecemeal, compiling in several passes, much as has been previously described for the 1620 - except that the compiler was on tape, and the intermediate output went to tape ... so you needed several tape drives to compile.) Inspired by FORGO, the need for a fast, in-core, student-oriented, load-and-go compiler with good error diagnostics was evident to lecturer Peter Shantz, and (under his direction) 4 third-year math students were selected to write the compiler during the summer of 1965: Richard Shirley, Angus German, James Mitchell and Bob Zarnke.[j]

«*Dave Sprott (first Dean of Math) says he deserves some credit for WATFOR. "I taught these guys statistics ... my lectures drove them into computer science."* **»**

Along with the FORTRAN and machine language codes, the compiler had to fit into a 32K word core memory (each having 36 bits), but by mid-summer, 1965, the WATFOR[k] compiler was completed. Compile times were five to fifty times faster than other available compilers and by the Fall it was in use at over 20 universities and computing centres in North America ... and (although IBM-Toronto yawned) the accolades came pouring in:

"... no way we could teach computing without such a compiler."

"This computing centre will never be the same again."

"Our users are astounded at the speed with which WATFOR compiles."

"... this batch of jobs, which took only a few minutes to compile, would have required the entire 24 hour day had we been using any other available FORTRAN IV compiler."

Within a year it would be adopted by computing centres in over eight countries, and the number of student users at UW increased to over 2500.

In July, 1966, the *Globe and Mail* reported:

Wes Graham, the high priest of the Canadian computer cult ... looks like a tweedy version of the Shah of Iran ... president of the 1000-member

---

[j] Someone recalls: "Gus spent an inordinate amount of time fixing his Austin Healey toy ...you wind it up - and it breaks down."

[k] Peter Shantz coined the name (sometimes written WhatFor?). I was told that Wes hated it. (PJP)

Computer Society of Canada.

The university's giant 7040 computer, a large oblong machine, increased its speed 100 times (with the FORTRAN IV compiler, which, the Globe said) *substitutes for a human being to work out mathematical formulas into computer language. For instance, Z equals X plus Y would become 1011101101111.*

75% of the 4500 students at Waterloo will receive computer training ... about 300 students are enrolled in the computer science program ... in three or four years a graduate can expect to earn $10,000 a year.

... starting in September ... about 190 teachers will take courses at the University of Toronto and Waterloo to prepare them to teach computer science courses (in the high schools).

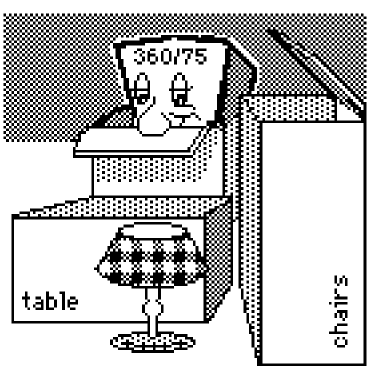
******

There never seemed to be enough money and cost-cutting measures were considered.

On the 1620, for example, Wes Graham noted that nearly all student programs used less than 40 columns of the 80-column card and had Don Cowan modify the software to use only the first 40 columns - so the cards could be read in again, backwards, and the last 40 columns used independently.

C.F. Arthur Beaumont (while Associate Dean of the Math Faculty, in charge of Faculty budgets among other things) recalls: *"Wes would come to my office, fill the blackboard with his ideas, then ask for money!"*

Yet, in spite of the lack of funds, there was a need for a larger, faster computer. Wes Graham had an idea. At the time, the provincial government was paying 90% of the cost of new buildings; the cost included furnishings.



Wes drove to Toronto to talk to an old undergraduate colleague: Bill Davis, then minister of education (and later, Premier of Ontario). At a meeting of the Computer Society of Canada in the Waterloo area, Bill Davis made the announcement:

The request for funds for a new Math and Computer building had been approved - and an IBM 360/75 was included in the request ... as "furnishings".

******

In early 1967, the 360/75 would arrive at UW, replacing the 7040-1401. It was the largest and fastest computer ever installed in Canada, with half a million bytes of core memory and an internal speed some 20 times that of the 7040. (Mutual Life Assurance company had also ordered a 360/75 and, until it arrived, they used the UW machine and expertise thereby establishing a close relationship with the university that continues to this day.)

Two lecturers in mathematics, Paul Dirksen (who was later to become the Director of the Department of Computing Services nee Computing Centre) and Paul Cress, were asked to lead a team to write WATFOR 360 ... and to begin *before the arrival of the 360/75* (a typical UW ritual!). The writing began in earnest during the summer of 1966, with team members working evenings each weekday then driving to IBM (Toronto), on Sunday, to check the code. Nothing worked.

By late summer, Wes had arranged for delivery of a 360 model 40[1] and within a few weeks the first FORTRAN statement was successfully compiled: *"I think it was something like 'x=a',"* Dirksen recalls with a smile.

"In the past, there seemed little need to write a fast compiler," says Wes Graham. *"For industrial applications - and that's where computers were being sold - you compile once, then use the program for a year. Nobody seemed concerned with educational needs ... with students waiting in line."*

WATFOR changed all that.

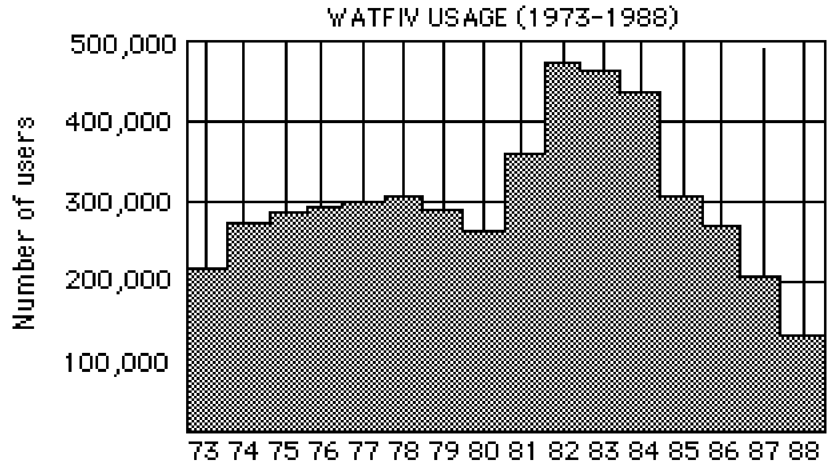
The *Financial Post* announced: *"Canadian linguists speed up computers"*.

WATFOR was announced at a "birds of a feather" meeting at a SHARE conference - and became an immediate success. It was distributed widely, initially free of charge, then a $300.00 one-time fee. (Eventually it went to an annual fee to provide the manpower required to supply updates/modifications.)

In 1972, Paul Dirksen and Paul Cress received the *Grace Murray Hopper Award* of the *ACM.* Soon, extensions to the FORTRAN language (the inclusion of CHARACTER types and, later, complete structured programming capabilities) were incorporated into yet another FORTRAN compiler: WATFIV.

---

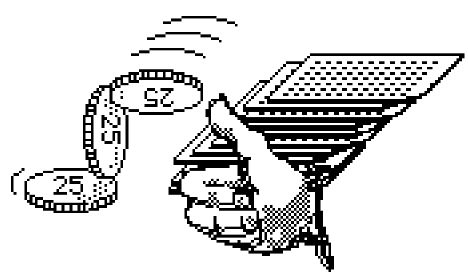
[1] Don Cowan recalls foam rubber attached to the 360/40 - so the machine could be kicked periodically.



By 1975, DATAPRO/DATAMATION surveys of software users had placed WATFIV in the top 25 of software packages surveyed.

Computer science changed rapidly as did the languages. Like FORTRAN IV, the available compilers were inadequate for the teaching of computer science and Waterloo had to write its own. By 1966/67, LISP was recognized as a language important to Artificial Intelligence, yet no accurate implementation of the language was available. Jerry Bolce (now with the Dept. of Computing Services) wrote a Waterloo version of LISP 1.5 (for the 360/75) and it was an immediate success. Stanford incorporated the code into their system and distributed it.

Joe Weizenbaum (of MIT) visited Waterloo in early 1968 and when he learned of the local implementation of LISP he challenged Wes Graham with a pet problem of his own (which, Joe said, could not be run on *any* LISP processor in existence). Wes won the bet (25¢ in U.S. funds).



******

In the Fall of 1964, the co-operative system of work-study was incorporated into the mathematics curriculum. *"We had planned on 40 students in our first class in co-op math, but had to take 100 due to the overwhelming response,"* said Ralph Stanton.

In January, 1965, "Computer Science Days" were instituted, with almost 4000 high school students visiting campus during the 1966/67 academic year, each student receiving a tour of the facilities, instruction in programming and three hours to write and run programs. By 1975, this number would grow to well over 10,000.

To teach elementary computer programming a simple language (a forerunner of BASIC) was devised for Computer Science Days: TUTOR. Paul Cress wrote the compiler.

And mathematics was growing. "It has been estimated that undergraduates majoring in mathematics increased, the world over, by 300% between 1956 and 1962". Waterloo was no exception ... except that mathematics was growing within a Faculty of Arts, and that created a problem.

"The need for greater visibility for mathematics and the growing importance of computing" (said Stanton), the requirement that curriculum in mathematics be established without the handicap of global Arts Faculty requirements and, in particular, the vision of Ralph Stanton, made inevitable the creation of a Faculty of Mathematics.

******

It was Fall, 1966.

Ralph Stanton and Ken Fryer were visiting high schools in the Lakehead (later to become Thunder Bay). These annual journeys had established a reputation for mathematics at Waterloo among the secondary schools of northern Ontario and were largely responsible for the early popularity of Waterloo among high school graduates. The typical *pitch* to the schools in the north emphasized the importance of a new area of study: Computer Science.

«Although Bev Marshman (Applied Math) recalls: "The attraction for me was simple: UW offered entrance scholarships to anyone who achieved 75% on the required grade 13 departmental exams; without that help I could never have gone to university." »

Ken and Ralph were always enthusiastically welcomed by the school administrators. Both had been active on the high school scene for years, with Stanton often setting the mathematics province-wide *departmental exams*. Ken's sense of humour helped:[m]

In Wawa: *we just came to take a gander at your goose.*

In Dryden: *every cloud has a sulphur lining*.

... and Ken had written a "whiffenspoof" song:

---

[m] High school students rarely understood Ralph's humour. Later, when some of us joined Ken on these northern tours, he would present a "crocopotamus award" for the best one-liner (having to do with the town we were about to visit). Ken always won!?*# (PJP)

From the high schools up in Cochrane
To the pool halls of Red Lake
From the stench of pulp and paper up in Dryden

We have heard about prime numbers
And the squaring of the cube
And the good life at U of Waterloo

Yes, the wonders of that good life
And the mathematics there
The computing, combinatorics, and the rest

We will head south on the morrow
And learn about WATFOR
Then we'll make a million dollars teaching you

But this day, in 1966, Ralph had to leave Ken and fly back to the University of Waterloo, early. Senate was meeting to vote on an important issue: a motion to establish a Faculty of Mathematics.

Ralph intended to be there.

On the plane, Ralph leaned back and listened to the hum of the engines. He pushed a pink tie into a worn jacket and glanced at his watch. There was time to relax, to reflect, to remember.

It was Ralph's most important, and most ambitious, undertaking. No other university on the continent had a Faculty of Mathematics. It had taken months of backroom bargaining. With the loss of mathematics, student enrolment in Arts would plummet. Most faculty members in Arts had expressed their opposition, yet, strangely, the chairmen in that Faculty had supported the idea. Perhaps they saw mathematics as an anomaly within Arts. Perhaps they saw it as a threat, the tail that wagged the dog, an abscess to be excised ... or perhaps it was simply the lobbying of the Chairman of Mathematics.[n]

Ralph remembered the day when Arts Faculty Council discussed the notion of a separate Faculty of Mathematics. The meeting was about to begin. Most members of Council had already taken their seats in the large amphitheatre. Then both doors at the back swung open, noisily, and all heads turned to watch. Ralph had rounded up every single member of the math department and lead them in gleeful procession to the meeting. Now they

---

[n] see Appendix A.

marched through the doors, two abreast, and filed into the back rows. Ralph sat by himself, several rows from the back wall so that all his people could see him. There was a low groan from the other members of the Arts Faculty. When it came time to vote on the issue, Ralph raised his hand. There was but a moment's pause, then every hand in the back rows rose in unison. Every hand, that is, but one. (That member of the department "sulked for a year then left".)

Stanton awoke as the plane descended to Toronto airport. He would be met by one of his graduate students: Jim G. Kalbfleisch (later to become Chairman of Statistics, then Dean of Mathematics, then Associate Provost of the University). Ralph looked at his watch. Enough time. The debate in senate would be almost enjoyable. He had done his job and there was little he could do now but wait - and listen to the discussion - and hope.

The opposition by Pete McBryde, Dean of Science, was to be expected, as was that of H. B. N. Hynes, Chairman of Biology. In April, 1965, Hynes had chaired a subcommittee of Science Faculty Council which reported that *Inasmuch as ... serious consideration is being given to the establishment of a Faculty of Mathematics, this committee recommends that the Science Faculty Council pass a resolution deploring the projected developments ... in the Department of Mathematics ...*[o]

Yet there *was* support within Science: Ian Dagg of Physics and W. F. "Bill" Forbes of Chemistry (who later joined the Math Faculty and became its second Dean). "Archie" N. Sherbourne, Dean of Engineering, led the unanimous support from that Faculty.

After the meeting, Ralph was pleased. As he wandered across the parking lot with hands stuffed into baggy pants, pink tie fluttering in a cool breeze, he couldn't help but smile. He would drive to his house in Doon and relax.

His Faculty of Mathematics had been approved.

Ralph grinned. Pete McBryde was waiting for him in the parking lot.
"I hope you didn't take offense to my opposition."
"Not at all," Ralph said. "I'm just happy it turned out the way it did."
"Well," replied the Dean of Science, "there was never any doubt, was

---

[o] *"Prof. Hynes' opposition to an idea often garnered backlash support,"* notes Ralph Stanton.

there?"

******

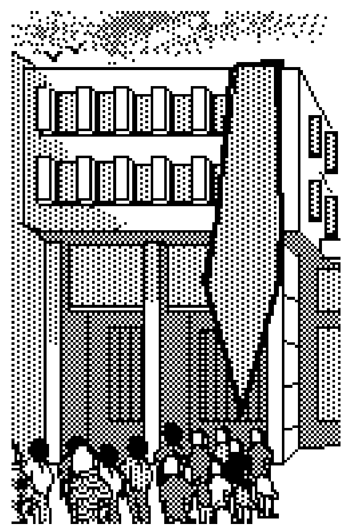
In 1966, the computing division within the math department had six[p] members: Wes Graham, Don Cowan, Jan Kent (who, at Don Cowan's invitation, joined the division from Norway and later went to Stanford), Paul Cress and Paul Dirksen ... and interest in computing at Waterloo had become widespread. Early relations were established with the Catholic University of Brazil, the University of Buffalo, and Nigeria.

In 1966, Ralph Gordon Stanton left the University of Waterloo. He had accomplished what he had set out to do. There were surely new challenges elsewhere. After spending three years at York University he became Chairman of Computer Science at the University of Manitoba and responsible for establishing a department of Applied Mathematics there.

Perhaps he hoped to repeat the miracle of Waterloo.

It didn't quite happen.

When the Math and Computer building was (unofficially) opened in 1967, the students celebrated with one of Ralph's gaudy pink ties ... and do, to this day.



«*Forbes Burkowski (currently a member of the CS dept.) has a theory about the colour of Ralph's ties. While a student at UW, he and a student friend, Bill Babichuk, lived with Ralph in his Doon house. Bill complained that, while washing his shorts, they turned pink - only to discover that Ralph's laundry water was runoff from a red roof.*»

Before he left, Ralph had hand-picked the first administrators of the new Faculty. He had envisioned five departments and, as harbingers of the new Faculty configuration, had earlier created five *divisions* within the math department:

---

[p] Earlier, Rocco L. Martino was a math prof teaching CS. "He gave his lectures lying on his back."

Donald D. Cowan would become Chairman of Computer Science,
Peter J. Ponzo would chair Applied Mathematics,
Gerald Berman would chair Operations Research,
Douglas G. Wertheim would chair Pure Mathematics, and
David A. Sprott would chair the department of Statistics.

Although Ralph had expected David Sprott to assume the Deanship, there was a "formal procedure" to be followed. George Cross (Pure Math) and David Sprott were nominated, there was a vote by members of the Math Faculty, and Dave lost - becoming the first Dean of Math.[q]

On January 1, 1967, the Faculty of Mathematics became operative, with Dave Sprott as Dean and Ken Fryer and Arthur Beaumont as Associate Deans (with graduate studies still in the hands of a University Dean, a position previously held by Stanton).

It was remarked at the time:
"When Ralph left, it took two Deans, two Associate Deans and five department Chairmen to replace him."[r]

Yet, in 1966, Senate had originally approved a Faculty of Mathematics with only *four* departments! Subsequently, various Faculty documents mention a "Department of Operations Research", then a "Department of Mathematical Science" which would "serve as a spawning ground for future departments such as Actuarial Science, Econometrics and Biometrics".

Trevor Boyes (UW Registrar, then and now), writes a memo to J.G. Hagey (Dec/66) reminding the president that :

"During the (Senate) *discussion it was made clear that only four departments had been approved ... I note that the recently published Faculty of Mathematics brochure lists five departments: Applied Mathematics, Computer Science, Mathematical Science, Pure Mathematics, and*

---

[q] There were suggestions made: "mark your ballot with a Cross", then "mark your ballot with a Sprott".

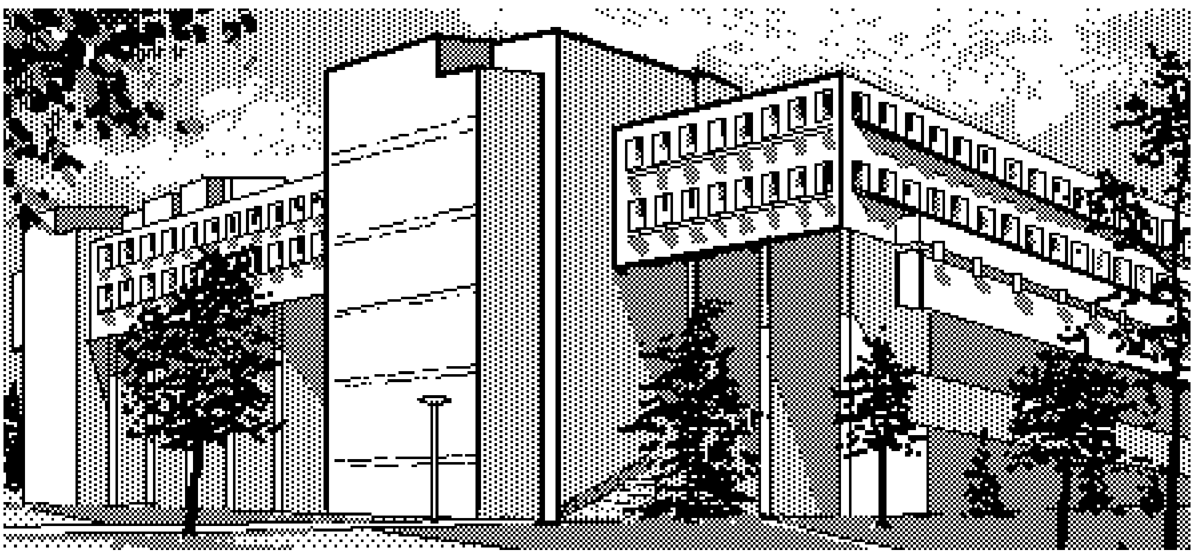
[r] Unfortunately, knowledge of the originator of this phrase had been lost to posterity, but this sentiment was widely quoted across the hallowed halls of the Math and Computer building. (It might also be pointed out that no cost-effectiveness studies were carried out to examine this comparison - which is probably best from the point of view of the latter group!)
... added just before the deadline for this CSH: Dave Sprott made the remark during the opening ceremonies of the Math and Computer building. (PJP)

Statistics."

Eventually, the errant fifth department *was* approved: Mathematical Science, later changed to Combinatorics & Optimization in time for the September, 1967, meeting of Math Faculty Council.

- The Computing Centre began moving into the new building early in 1967 with its complement of equipment, including an IBM 1620, 1710, 7040 and 360/75. (Faculty members moved late in 1968.[s])

- By the Fall of 1967, there were expected to be over 1500 students enrolled in the Faculty.



The Mathematics and Computer Building

- The new Mathematics Faculty Council unanimously approved a motion to call the building *Stanton Hall* (although *Stanton Centre* and *Stanton Hall of Mathematics* had also been suggested).

  Unfortunately, it didn't happen.

---

[s] Bev Marshman (now in Applied Math) recalls wandering through the unfinished 5th floor picking out an office, later to discover that grad students were relegated to the 1st floor.

Chapter Two

the Early Years 1967 -1975

It was summer, 1966, and Don Cowan had just returned from a trip to Michigan. Wes Graham approached him in the hall.

"Ralph was looking for you ... wants to see you."

"Ralph? What about?"

Wes grinned, but said nothing.

Don stood uncomfortably in Ralph's office in the Physics and Mathematics building. He had been actively involved in the computing division within the math department, had received his PhD the year before, had assumed that his performance was acceptable.

What did Ralph want?

"I'd like you to head up the department's computing division and next year, when the Faculty is in operation, become the Chairman of the Computer Science department."

Less than two years from his PhD and he would be the founding Chairman of the Computer Science department; heady stuff. The smile was still on Don's face when Janos Aczel approached him with an offer he couldn't refuse.

Aczel had assembled a research group in Functional Equations and suggested that this group would be pleased to join the new department[a] - but the name should be changed to Applied Analysis and Computer Science. It *was* changed, and the department was born AA/CS.[x]

******

The 1967/68 UW calendar lists the founding members of the department of Applied Analysis and Computer Science:[b]

Professors:

Janos Aczel* PhD (Budapest), Habil.DSc (Hung. Ac. of Sc.)

---

[a] As the first chairman of the (empty) Applied Math department, I can recall running around asking people: "Next year, when we're a Faculty, wanna join AM, huh?" (PJP)

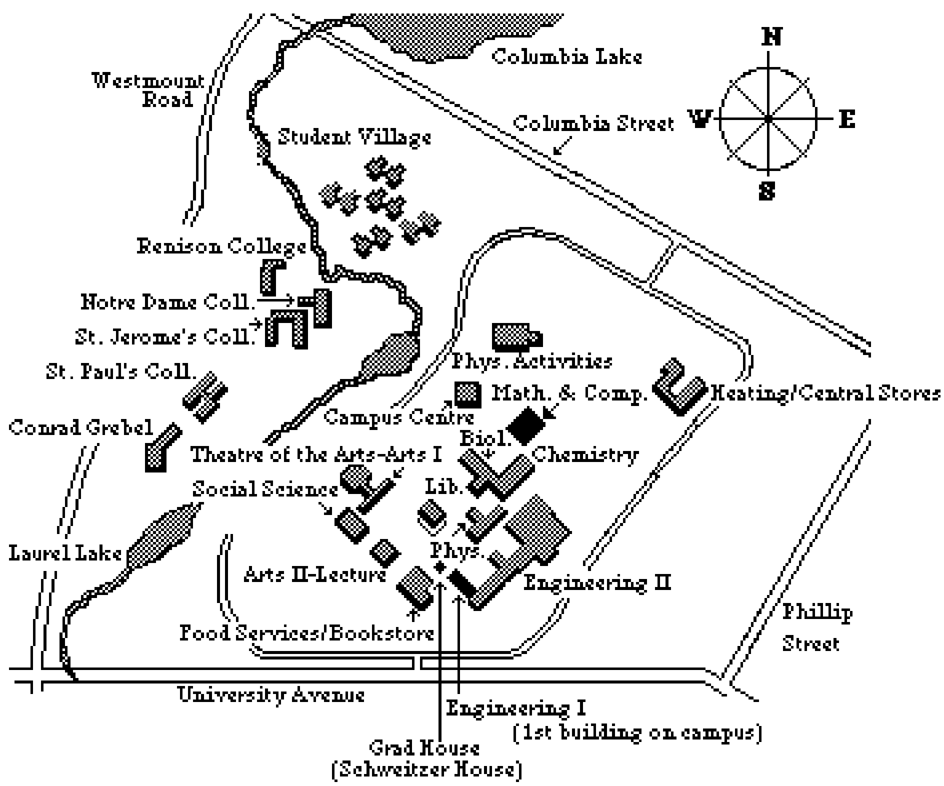
[x] Dave Sprott, Dean of Math, recalls the headaches associated with choosing a "suitable" name. He eventually got agreement on Applied Analysis and Computer Science.

[b] In the listing below I only identify where the *highest* degree was obtained. (PJP)

J. Wesley Graham BA, MA (Toronto) Director of Computing Centre
H. Haruki* PhD (Osaka)
Mike A. McKiernan* MA, PhD (IIT)

Associate Professors:
Donald D. Cowan BASc, MSc, PhD (Waterloo) AA/CS Dept. Chair
J. Douglas Lawson BASc, PhD (Waterloo)
Assistant Professors:
Peter C. Jennings Cross-appointment (Psychology)
John. C. Wilson BASc, PhD (Waterloo)
Lecturers:
Peter C. Brillinger BSc, MA (Waterloo)
Paul H. Cress BSc, MSc (Waterloo)
Paul H. Dirksen MSc (Waterloo)
Jan G. Kent MSc (Oslo)
Byron H. Ehle BA, MS (Stanford)

* (members of the Functional Equations group)

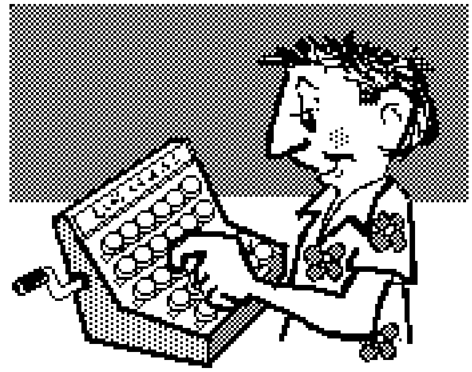


Campus Map : 1967

There were few courses in this new academic area: *computer science*.

The original first-year course, Math 132 as devised by Ralph Stanton, was described in the 1967/68 calendar much as it was in the 1966/67 calendar:

Math 132.
Numerical Procedures. *Introduction to finite differences. Interpolation. Numerical integration. Introduction to the use of electronic computers and electric calculators.*



The 1967/68 requirements for the Honours degree in AA/CS were little different from the requirements for any other department:

Math 130 (Calculus), 131(Algebra), 132(Numerical Procedures) + 3 electives.

Math 230 (Algebra), 233 (Probability/Statistics), 237(Calculus), either 236 (Diff. Eqns) or 238 (Principles of Computer Science) + 3 electives.

Math 331 (Algebra), 332 (Complex Variables), 334 (Numerical Methods), 340 (Computer Systems), either 333 (Diff. Eqns) or 338 (Statistics) + 2 electives.

Math 431 (Algebra), 436 (Logic), 427 (Graph Theory) + 2 other maths + 2 electives.

making a total of 27 full-year courses (each worth *1 credit*). Although co-op math was introduced in 1964, most courses were still *full-year courses*, the first and second halves of the course (each worth *.5 credits*) being separated by a co-op workterm, the final grade being a weighted average of the two marks. If a student failed a final exam a "supplemental exam" was written. Students who failed too many courses repeated the entire year (including passed courses). The current *credit system* would be introduced years later.

Math 238 and Math 340 are described:

Math 238. Principles of Computer Science. *Characteristics and design of digital computing machines. Programming languages. Algorithms and flow diagrams. Design of assembly systems, interpreters and compilers. Advanced programming techniques and operating systems.*

Math 340. Computer Systems. *A discussion of the hardware and software required in a computer system. Overlapped channels, interrupt facilities, memory protection, buffers, input-output control systems, macro-programming, monitor systems. Relocation schemes, multi-programming, multi-processing, dynamic memory allocation, time-sharing. Special purpose computer systems and simulation of computer systems. Peripheral equipment.*

Introduction to the theory of sequential machines and logical design of computers.

1967/68 enrolments in the new faculty were:

Regular program	*700*
Co-op program	*600*
Graduate program	*130*
TOTAL	*1,430*

Computer Science was rapidly becoming the most popular program within the Faculty (and, along with Actuarial Science, the only co-op program.)[c]

There were, by 1967, more than 150 companies employing co-op math students, with salaries averaging $350.00 per month for a first-year co-op student. For graduates, "salaries in industry vary widely, but certainly range from $6000[d] to well over $20,000."

In 1967, William G. Davis, Ontario's Minister of Education, predicted that children would teach themselves to read by means of "talking typewriters" in schools administered not by a staff of teachers, but by a vast complex of electronic devices.

******

For Don Cowan, in 1967, the task at hand was to find good people in computer science, to design an expanded curriculum, to establish a graduate program. John Brzozowski (later to become the fourth Chairman of CS) wrote Don, visited Waterloo to give a seminar and joined the fledgling department in 1967, its first year of operation. John was soon to be followed, in Spring, 1968, by Eric Manning, who would eventually become founding Director of *both* the Computer Communication Network Group (CCNG) and the Institute for Computer Research (ICR) - see elsewhere in this CSH for more details.[e] Then cameTom Pietrzykowski, Charlotte and Pat Fischer (Pat

---

[c] Wes Graham insists that *he* first proposed "co-op math/CS". Ralph says it was *his* idea. In any case, Wes worked with the dept. of coordination, running around signing up co-op employers.

[d] A student, Robert Dickson, comments: "Today (1992), a coop student wouldn't accept that for a 4-month work-term."

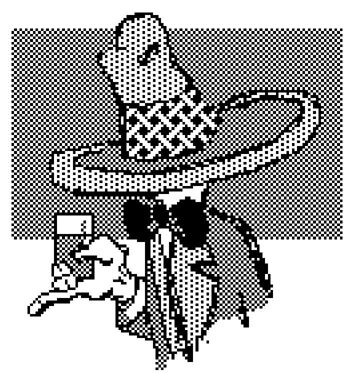
[e] It would be said of Manning: *"Eric made a tremendous contribution through his liaisons with the federal government, the EE dept, and industry (in a very large sense), coupled with a grand, personal vision of the role of CS/EE at Waterloo, and truly remarkable public relations skills. Among other things, he was a superb enabler who is undisputably responsible for a huge influx of research funding and opportunities."* Bruce Simpson (CS Chair, 1984-1987) has characterized Eric as *"one of several creative builders of almost heroic proportions"*. As though to confirm these comments, Eric eventually became Dean of Engineering at the University of Victoria where, in Eric's words: *"We've built an Engineering Faculty which is unique in Canada, in that almost all of it is devoted to Information Technology. When I came we had 8 faculty members in 1 department, about 100 undergrads and about 20 postgrads. We now have 60 faculty in 3 departments (Computer Science, Electrical & Computer Engineering, and Mechanical Engineering), 650 undergraduates and 200 postgraduate students. We won nodes of three of the 14 Federal networks of Centers of Excellence and place our co-op students nationwide and overseas.*

was later to become the second Chairman of AA/CS) and Jim Linders. (Jim later became Chairman of Computer Science at the University of Guelph.)

John and Pat worked to develop theory courses in computing and Don grew fat entertaining prospective faculty. With Eric Manning, he toured the U.S. universities on the East and West coasts. During a Berkeley/Stanford/UCLA tour they found Alan George (who joined the department in 1970 and was later to become the third Dean of the Math Faculty, then Academic VP and Provost) and Johnnie Wong (who would later become UW's Associate Provost, Computing and Information Systems).

- IBM typically sponsored visits, to UW, of professors and other dignitaries from around the world.

  "When the Mexicans came, we wore big hats and had to learn the proper way to drink *tequila*," says Wes Graham.



- By early 1967 the *Brazil Connection* was firmly established:

  In 1966 a group from the Catholic University of Brazil visited UW; among the group was Carlos Lucena.[f] He encouraged IBM to bring the UW expertise to South America. In 1967, at IBM's request (and expense) Wes Graham toured South America for some six weeks talking about computing and what Waterloo was doing. Soon after, the new Liberal government of Canada sent its minister for external affairs (M. Sharp) on a diplomatic tour of South America; he found that "everybody was talking about Waterloo". When he returned he phoned Wes "to find out what was going on". (The dept. of external affairs subsequently sent other UW faculty and staff to South America). Later, Don Cowan and Archie Sherbourne (Dean of Engineering) became involved ... and the *Brazil Connection* continues to this day with

---

We've built one new building - an office tower - and are now designing our second new building - a lab wing, which will be about 0.35 DC in size (0.35 of a Davis Center; the unit I use to measure such things!)"

So why did Eric leave UW? Aaah ... that's another story!

[f] I met Carlos while I was writing this CSH; he was visiting UW! (PJP)

graduate students and visiting faculty. (While Wes was in South America, Don Cowan was "in charge" and was pressured into giving Engineering $100K to buy a PDP-9 (?!) from the $1.2 million computer equipment grant for the new math building. Wes was "delighted". (*sic*))

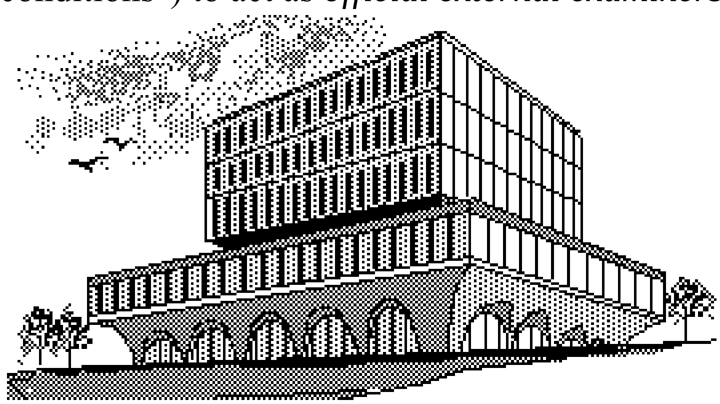
- As with most departments in the Faculty, much of the teaching was done by graduate student *lecturers*. It was a technique used by Ralph Stanton and continued successfully for years after the creation of the Faculty (but has since fallen out of favour). Many of the graduates went on to become faculty members at Waterloo and elsewhere (and, indeed, accounted for a few Chairs, Deans and Associate Provosts).

- Don tried unsuccessfully to lure Jim Mitchell back to Waterloo. (Mitchell, along with Richard Shirley, Gus German and Bob Zarnke, authored WATFOR.)

- For two years, AA/CS faculty flew to the University of Lagos (living in "less than admirable conditions") to act as *official external examiners.*

Arts Library, 1967:

There were rumours that the library wasn't designed to support the weight of books (or an additional 3 storeys).



- The demands of computer science education made the writing, maintaining and distribution of software a major activity. A technical staff was assembled, later to become the Computer Systems Group (now directed by Eric Mackie). CSG became the natural successor to the teams which wrote the WATFOR and WATFIV compilers. When some introductory courses switched from FORTRAN to COBOL (a switch necessitated by the introduction of co-op computer science ... "employers demanded COBOL"), CSG wrote the WATBOL compiler. Later they distributed a text formatting package, Waterloo SCRIPT, written by the Department of Computing Services (DCS). By 1982, the "25th anniversary issue" of the DCS newsletter noted that there were 420 institutions using WATFIV, 230 using WATBOL and 370 using

DCS's SCRIPT.[g]

******

In 1968, the Sunday Times (London, England) reported:

Cafeteria-style service is the order of the day in the computer science courses at Canada's University of Waterloo.

In many ways Waterloo leads the world in developing the methods of mass education for the computer age.

Waterloo's 360/75 is now processing 2,000 student programs every day which take on the average only about half a second to compile and execute and mop up only 10 per cent of the machine's time, the rest being used for research and administration.

Until computer cafeterias are a regular feature of Britain's schools and universities they will not be providing the education that will be needed in the future.

A reporter for the Daily Telegraph (London, England) noted:

Waterloo is pioneering a system which will undoubtedly spread to other universities and colleges.

The university's concept of computer involvement extends into the high schools. Thousands tour the complex each year while the prize for the top 50 high school maths students is a week's course at Waterloo. A 14-year-old boy was handing his correctly programmed problem to the computer after only one lecture on techniques during my visit.

The preferred text for WATFOR and WATFIV was, naturally, written at Waterloo (by Cress, Dirksen and Graham).

******

By 1968 the calendar description for Math 132 had changed:

Math 132. <u>Introduction to Computer Science</u>. *A thorough introduction to algorithms, stored programme computers and programming languages. Concepts and properties of an algorithm, language and notation for describing algorithms. Analysis of computational problems and development of algorithms for their solution. A procedure-oriented language (FORTRAN IV) and machine and assembly languages are used to implement algorithms on the computer.*

A hypothetical computer called SPECTRE was introduced to accompany the machine language/assembler component of this course. The

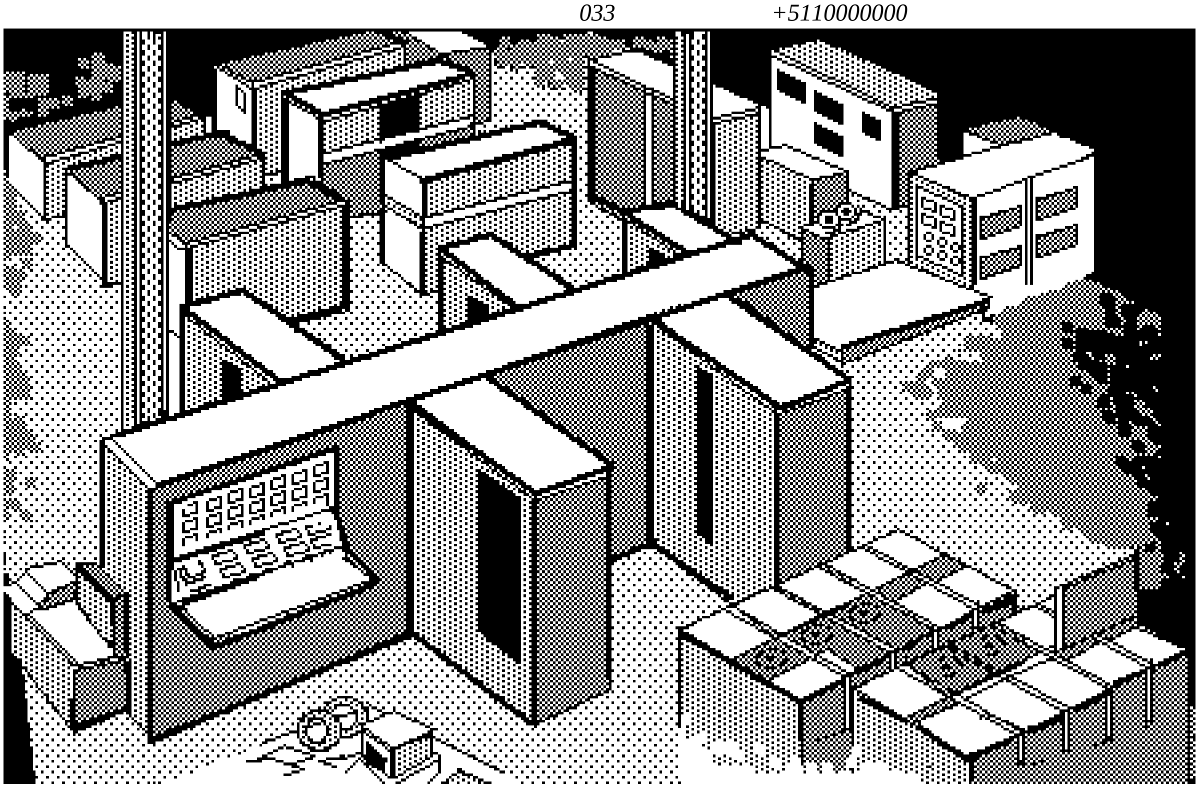
[g] see Appendix C for a more complete history of CSG.

original SPECTRE simulation was written in 1965, in assembler for the 7040, by Wes Graham, Bob Zarnke and Gus German, with a "SPECTRE and SPECTRE MAP User's Guide" written by John Vellinga. The successor to SPECTRE was another hypothetical computer, WATIAC (with WATMAP assembler), designed in 1973 to teach the basics of machine/assembly language. The textbook was published by a new organization: the Waterloo Foundation for the Advancement of Computing (WATFAC). Various modifications were made to SPECTRE; the final program in the 1968 text (by Peter Brillinger, Byron Ehle and Wes Graham) begins:

		Location	Instruction
BEGIN	*RN2 X*	*000*	*+0000072012*
	LDQ X	*001*	*+0000012012*
	TSL SIN	*002*	*+0000056016*

and ends:

ARG		*029*	*(saved for ARG)*
ARG2		*030*	*(saved for ARG2)*
	END BEGIN	*031*	*+4883333333*
		032	*+5016666666*
		033	*+5110000000*



The 360/75 moved into the "red room" months before faculty moved into the building[h]

• By 1969 the AA/CS department was the largest in the Faculty with

[h] Don Cowan wandered in with son John in arms. John burped, punched a button, and brought the greatest machine in the country to its knees.

research areas in Computer Graphics, Formal Languages, Switching Theory, Theory of Automata, Numerical Analysis, Information Theory, Functional Equations, Compiler Construction, Programming Systems, Machine Organization and Fault Diagnosis. Until 1969, Pure Math had been the largest department, consisting of most of the original members of the math department (when it lived within the Faculty of Arts).

- Also, in 1969, the first students from grade 1 (six- and seven-year olds from Rockway school) visited Waterloo to learn about the high-tech world of computers. *"They were impressed and interested and learned the basic concepts. Now we'll collect cardboard boxes so we can build our own,"* said their teacher.

- By 1970 the Math Faculty had dropped the "progress by year" philosophy and adopted a "credit system". Later, all other Faculties (except Engineering) would follow their lead.

- In 1972, the first IBM 370 was installed (model 145), to be replaced a few years later by a 370/158. (By 1981, Hewlett-Packard would announce that processing power equivalent to a 370/168 was available on a single chip!)

- In the same year, Mathematics Faculty Council expressed "sturdy support" for a motion (by Don Cowan and Pat Fischer) to change its name to the Faculty of Mathematical Sciences. It didn't happen. The Science Faculty Council objected "in the strongest possible terms" and immediately voted in favour of a motion to change *its* name to the Faculty of Mathematical and Experimental Sciences.

  It was not the last time such a motion would be put to Council. In March, 1992, the suggested name would be "Faculty of Mathematical and Computer Sciences". Students (and others) objected, a vote was taken and the motion was defeated (in April/92, just before this CSH went to press).

- Pete Brillinger and Don Cowan asked Faculty Council to approve the appending of the phrase "Honours Applied Analysis and Computer Science - Co-operative Program" to the B.Math degree in Computing. That didn't happen either.

- A committee was struck to decide whether a room with computer terminals would be placed on the fifth floor of the Math and Computer building. The Pure Math department boycotted the committee (*"too much money spent on computer equipment"*, *"too noisy"*, etc.). The terminals *were* installed, nevertheless, and the first user was the Chairman of Pure Math.

- The Computer Communications Network Group (CCNG), a joint venture between Electrical Engineering and Computer Science, was established in 1972 with Eric Manning as Director. The following year it received a $750,000 grant from the National Research Council and, among other things, worked on Bell Canada's Datapac.

«*Although $750K was forthcoming, space wasn't. Manning noted that unless space was found the grant would have to be returned. In the 11th hour, space was found in Engineering 4 (Carl Pollock Hall). At the time, Engineering and Math seemed to regard each other as foreign powers, probably hostile. One day Eric Manning was walking down the hall in CPH. An Engineering colleague stopped him to enquire what he (a "mathie") was doing THERE? Things *have definitely* improvedbetween these Faculties!* »

By 1978 the Financial Post was to write:

The worldwide reputation of Canada (in computer/communications) *... some of it comes from the work of the CCNG at Ontario's University of Waterloo. The group is in the forefront of theoretical research ...*

- In the Fall of 1973, Wes Graham resigned as Director of the Computing Centre to be replaced by Paul Dirksen (a position he still holds).

******

Pat Fischer met Charlotte Froese while he was at Cornell University and she was at the University of British Columbia. UBC wouldn't give Charlotte a leave of absence so Pat went to UBC as a visitor. They married in 1967, looked around for a place where they could both work, noted that positions were available at Waterloo and joined the AA/CS department in 1968 (with Charlotte Froese-Fischer holding a joint appointment in Applied Math). When Don Cowan's five-year term as Chairman ended, Pat was on the selection committee for a new Chairman. He resigned, ran for the position himself, and became the department's second Chairman in 1972.

******

In the early 70's it was time for the Math Faculty to provide mainframe computing to support the special needs of its students and faculty ... and the Math Faculty Computing Facility (MFCF) was established with Morven Gentleman as its first Director. The first machine purchased was an IBM series 50, but a more sophisticated computer was necessary. A committee was struck to decide; the committee included Pat Fischer, Morven Gentleman, Wes Graham and Jim Linders. They met several times and "agreed": the Faculty should purchase a PDP-10 (Readers familiar with the personalities of these individuals will know that these meetings took place in a calm, detached and friendly manner in which everyone tried to learn from each other, and that the results of these scholarly deliberations resulted in agreement based upon sound knowledge.)

Poof ... in September, 1972, a Honeywell 6050 (the *Honeybun*) was installed!

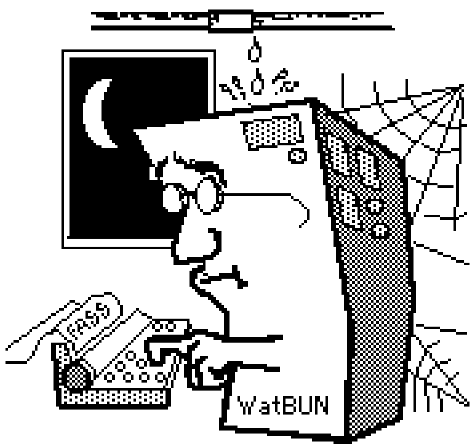
«*Bill Forbes, Dean of Math at the time, recalls: "It was an interesting time. Both Digital and Honeywell approached us and their initial pitch was that they were concerned that the thousands of students taking courses in Computer Science and Mathematics were predominantly involved with IBM equipment, and they would like to ensure that they were also exposed to their equipment. This was followed by statements to the effect that they would make us offers which we could not refuse. This, in turn, was followed by progressively better offers from both manufacturers until it became clear that they, in fact, represented offers which could not be refused."* »

Honeywell had been eager to enhance its presence on campus after introducing, in 1971, its new 6000 series computers ... and MFCF wanted a timesharing machine (the first *general purspose* machine on campus[i]). The *bun* became the Faculty workhorse, processing over 6000 jobs and printing well over 15 million lines each month ... and providing the first UW electronic mail service. Students flocked to the machine and wrote software and a "UW Software Tools Package" was generated (becoming standard on all *bun*-like machines). In 1974, Honeywell provided title to $250,000 worth of auxiliary equipment - and faculty in the AA/CS department provided a specially-developed version of APL which "substantially exceeds specifications". In 1975 the 6050 capacity was increased to that of a Honeywell 66/60, then again in 1981 to that of a 66/80. In 1983, the 66/80 was replaced by a Honeywell DPS8/49, adequate to support peak loads of 90 simultaneous users and provide on-line file storage for over 5000 users.

---

[i] For several years before the Honeywell, two IBM 650's had been servicing 60 terminals running APL.

Students were *encouraged* to crash the system in order to identify any weaknesses. Gary Embro (MFCF) recalls: *"Anyone caught crashing the system and could demonstrate such to us was taken to St. Agatha and treated to a jug of draft."*



Alas, mini- and micro-computers came on the scene (a typical PC now providing more memory, disk storage and speed than the original *bun*) as well as a host of Unix machines (the first DEC VAX 11/780 arriving in 1980). Usage shifted away from the *bun*. The poor *bun* still ran off the occasional FASS[j] script ("the world's most expensive copier") and provided access to the Adventure game and Wumpus ... but the end was in sight for the DPS8/49.

On April Fool's Day, 1992, the poor old *bun* was put to rest ... but memories remain of ruptured plumbing and sludge-on-a-bun and late night hackers rinsing circuit boards in the washroom and drying them under the hot air hand driers.

«Above the Honeywell was Bill Forbes' lab; someone turned on the H_2O before the drains were installed.»

******

In 1970 the *Kitchener-Waterloo Record* noted:

"... economists predict that, by 1980, Canadians will be spending more on computers than on cars"

"... a computer service will be available to calculate income taxes."

In 1971 the IBM 360/75 finally answered the age-old question "which came first, the chicken or the egg."

Answer? The egg.[k]

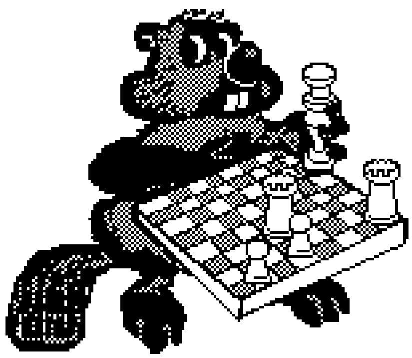
---

[j] "Amateur night", by Faculty, Administration, Staff and Students (created, I think, by Ken Fryer). (PJP)

[k] The question, posed by the Ontario Egg Marketing Board: "If a chicken started 10 feet behind an egg but flew 1 1/2 times as fast as the egg rolled, and the finish line was 28 feet from where the chicken started, which came first?"

Courses continued to evolve, reflecting the changing scene in computer science. A Sept/73 issue of mathNEWS[1] notes:

> • *Anyone who remembers Math 132 (now A and B), will be pleased to know that FORTRAN is no longer the major first-year language. Unfortunately, COBOL is.*
> • *Math 132B should be fun too. After playing around with SPECTRE MAP ... a new thing, alias WATMAP, which should go well with Math 132A since it is written in COBOL (what else would you simulate an assembly language in?)*
> • *Math 240B (which, as we all know, comes before Math 240A) has also been given a face-lifting. You now learn ALGOLW (... most emphatically not the same as ALGOL) and SNOBOL (no comment) and L6 (supposedly a list processor). If anyone gets through all that they'll learn PL/1 as a filler.*



In 1974, a UW program, *Ribbit* (written by CS students, including Ron Hansen, Russell Crook and Jim Parry) won the U.S. Computer Chess championship and placed third in the world championship.

[1] the student newspaper.

Chapter Three

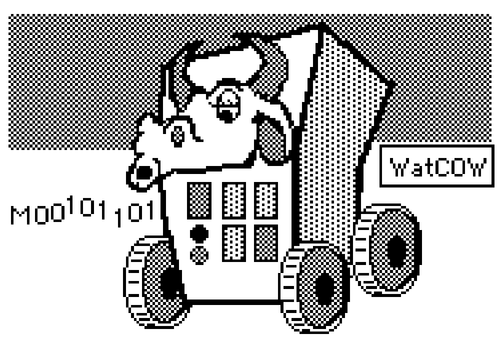
Modern Times 1975 -1992

By the time J. Douglas Lawson replaced Pat Fischer as Chairman (1975), the name of the department had reverted to Computer Science and boasted 8 full professors, 7 associate professors, 14 assistant professors, 5 research assistant professors, 4 lecturers, a postdoctoral fellow and 2 adjunct professors ... and Math 132A/B still taught COBOL and machine/assembly language.

- Math Faculty Council approved a renaming of all math courses: Math 132A/B became CS 180 and CS 140. There were now 37 undergraduate courses with CS prefix.[a] (Within five years, the department would be offering 46 undergraduate courses.) About half of the undergraduate enrolment could be identified as *computer science* students. Other departments were beginning to jump on the CS bandwagon with programs *Actuarial Science and Computer Science*, *Statistics and Computer Science* and, later, *Pure Math and Computer Science* and *Applied Math and Computer Science*.

- In 1975 the WIDJET system (compliments of the Computer Systems Group[b]) was introduced on PDP-11's. Gone forever were paper tape and punched cards. Students used terminals to create and run jobs (and reduce the amount of paper output!). By 1977 it would be running more than 5,000 jobs per day.

  CSG designed and built a "portable computer", WatCOW (a Computer On Wheels) which was taken on demonstration tours and programmed on-the-spot.



[a] see Appendix B.
[b] see Appendix C

- In 1978, "UW's pride and joy", the Honeywell was hooked into Bell Canada's computer network, Datapac (then serving 72 cities).
- The Waterloo Region Review Committee noted that *"... this year more than $7 million will be spent on computers and data processing by public organizations ..."*
- Under the heading *"Cities waste millions on computers",* the *K-W Record* comments on the committee's report:*"... maybe UW computing should be part of a public facility"*
- A year later (1979) the once formidable IBM 360/75 was put up for auction.

«*It wasn't easy to get rid of the 360/75. George Hill (DCS) tells the following story. IBM was hosting some group at the Waterloo Motor Inn and they were all standing around the indoor swimming pool. George had hired a voluptuous undergraduate; she walked in, slipped off her robe, dived into the pool (with all eyes watching) then climbed out and left. A sign on the back of her robe read: "For Sale ... IBM 360/75"* »

******

The house was locked, so John and Maria Brzozowski peered through the window. They had come to Waterloo in the Summer of 1967, had seen this house by day and returned in the evening with Wes Graham, seeking his advice.

But it was locked.

******

Janusz A. "John" Brzozowski had obtained his PhD at Princeton in 1962 then returned to Canada, to the E.E. department at the University of Ottawa, hoping to establish a group in digital system theory. It was happening, but slowly.

Then he heard of Waterloo: *"They had buildings, computers, students, money ... and they needed to build up their faculty. The prospects seemed good. Both Don Cowan and Wes Graham met me at the airport ... pretty impressive."*

******

John would join the new AA/CS department in the Fall of 1967.

But now, in the Summer of '67, he and his wife were looking for a house, peering through the window.

Wes Graham looked through the window.

"Hey, look at that chandelier! Gotta be worth at least $200.^{00}. I'd say this was a good buy!"

So they bought it - and it *was* a good buy - and the Brzozowski's still live there!

John hoped to build up the theoretical side of computer science at UW. The department was actively recruiting and there was an opportunity to attract good people, but it was important to create an image *outside* UW, in theoretical computer science.

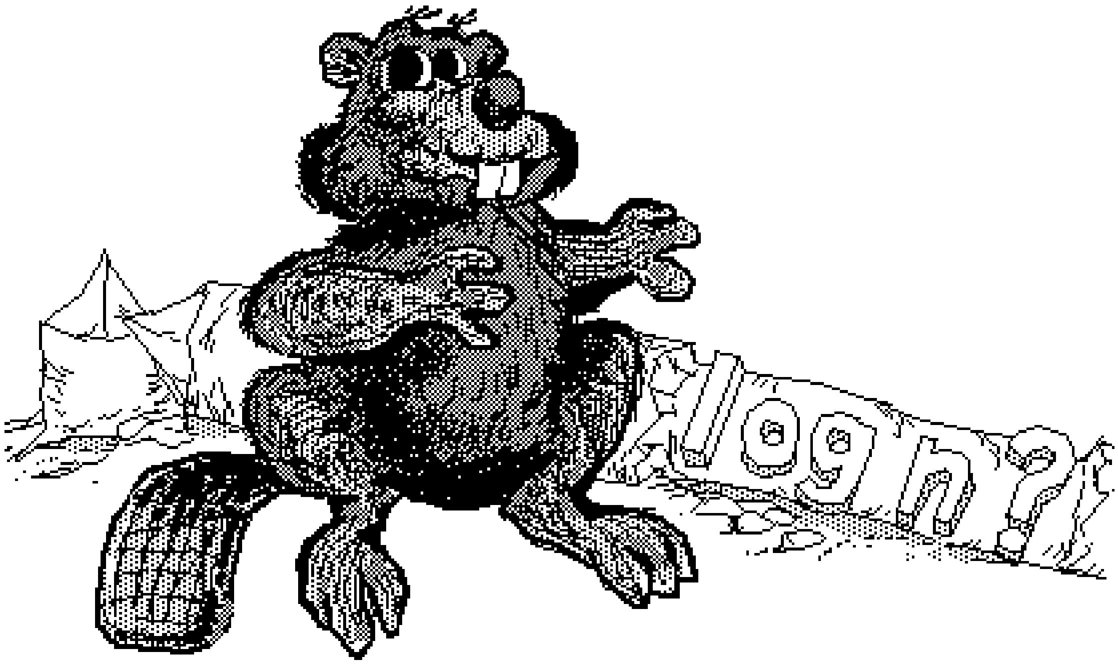
The opportunity came soon. The Switching and Automata Theory Committee of the IEEE Computer Group was approached; they would hold the *Tenth Annual Symposium on Switching and Automata Theory* at Waterloo,[c] in October, 1969. It was the first time the symposium would be held outside of the U.S. and it was a golden opportunity for the young AA/CS department - *if successful*. A call for papers went out and the response was gratifying: over two dozen papers presented by many distinguished researchers in the field (including A.V. Aho, J.E. Hopcroft J.D. Ullman).

Over the next few years the *Theory Group* grew in strength. By 1977, Ashcroft, Brzozowski, Culik, Munro and Pietrzykowski could organize a conference on their own: the *Conference on Theoretical Computer Science*, featuring three dozen papers by leaders in the field (including a paper co-authored by R.E. Tarjan).

Ed Ashcroft designed a conference logo[d] to reflect the image of the "busy beaver problem" (concerning the longest output generated by a finite state Turing machine):

[c] SWATerloo was the local euphemism.

[d] See Appendix J for other mascots/logos/etc.



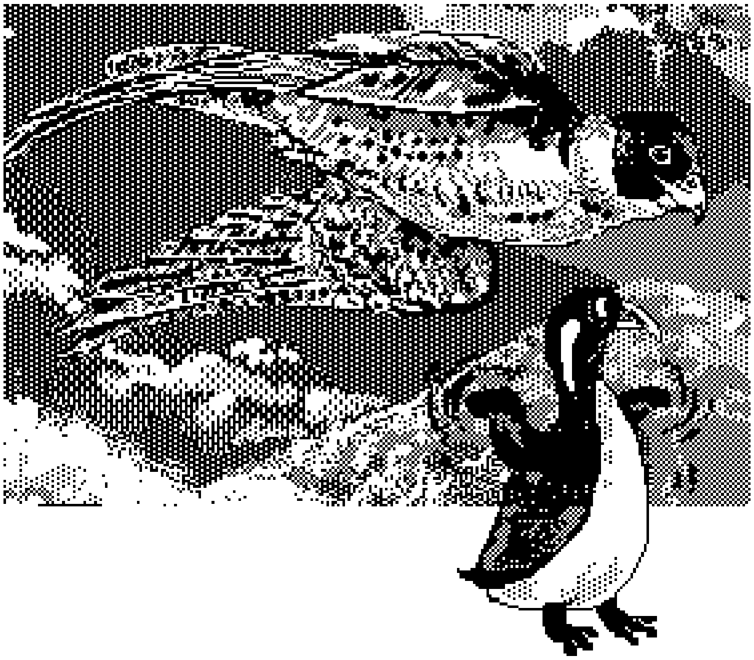
At a subsequent CS department meeting, Kelly Booth moved that the beaver be "embraced" as the department logo.[e]

And it was, and it is.

The mascot for the Artificial Intelligence Group, the "penguin", also arose in connection with a CS problem. Chrysanne DiMarco (in the CS dept.) notes that *"it is a classic AI example about the difficulties of representing facts in logic".*

---

[e] although Booth claims it was Brzozowski who made the motion!



if x is a bird,
then fly (x)
if y is a penguin,
then not (fly (y))
if x is a penguin,
then x is a bird

bird (tweety)
penguin (tweety)

Does Tweety fly??

«*When I asked CS faculty why the penguin was the* ***ai*** *mascot, Frank Tompa (with access to the Oxford Dictionary!) responded: "I thought the* ***ai*** *mascot was the* ***ai****." I pulled out my dictionary and found:* **ai** *n* A three-toed sloth of South America (*Bradypus tridactylus)* . Very funny, Frank. (PJP) »

******

In 1978, J. A. Brzozowski succeeded Doug Lawson as CS department Chairman. It would not be the last time he held this post. Indeed, it would not be the last time Lawson would chair the department.

******

Microcomputers (the TRS-80, Commodore PET, Apple) entered the computing arena in the late 1970's and promised access to low cost computing ... but memory was limited. (The original PET was an 8K machine with 40 column screen.) The Computer Systems Group designed the MicroWAT with bank-switched memory (originally up to 640K, soon to be expanded to a megabyte ... IBM bought one for $110,000 before their PC was introduced). The 32K Commodore 8032 was modified along MicroWAT lines and became the SuperPET. Until the IBM PC arrived on the scene (1982), UW supported courses with SuperPETs.

******

1982/83 were good years:

- The *Toronto Star* described Kelly Booth as a UW "computer artist".

- Hewlett-Packard announced a "computer partnership" with UW, and a gift of a new line of computing equipment worth $250,000 (featuring an HP 9836 and auxiliary equipment) to be used by Wes Graham, Keith Geddes, Gaston Gonnet, Bruce Simpson and Doug Lawson.

- IBM announced a $17.5 million research and development deal with UW, including $5 million in state-of-the-art equipment (including 120 personal computers) and over $1 million/year for five years to support the research. An IBM spokesman said: *"UW produces about one third of all computer science graduates ... their exposure to enhanced facilities will allow students to have a positive impact on Canadian industry ..."* The three recipients were Wes Graham (as Director of CSG), Eric Manning (as Director of ICR and Paul Dirksen (as Director of the Computing Centre). The *Kitchener-Waterloo Record* heralded the beginnings of a Silicon Valley North!

- The *Globe and Mail* writes, in an article called "the Campus where computer is king":
  Waterloo ... a young upstart ... with the emphasis on computer literacy.
  Waterloo has breached the tradition that says the oldest schools (Oxford, Harvard, University of Toronto) are the best schools, but some of the more established institutions seem to resent its style.
  (The math Chairman of U of Toronto says) *"... as far as quality of program, it's quite clear that U of T is number one at the graduate level."*
  This is an almost embarrassing gung-ho place. The academics do not act like academics; they behave more like a combination of evangelical socialists and profit-motivated capitalists. In two words: intense and committed.
  Waterloo does ... have a serious problem: finding and keeping top talent on staff ... computer scientists can be earning $45,000 or more a year.

«*There had always been a problem of retaining faculty members; they wandered in, then wandered out. One chairman noted: "Everyone has his favourite explanation; some faculty will say too much teaching. The dentists say it's due to not brushing teeth." During the 80's, the CS dept. lost perhaps 20 faculty. Someone remarked: " I think UW could benefit from a lot more care and skill spent on good human relations, a comfortable atmosphere - consultation, communication and simple civility. That was one* *of many things that Ken Fryer did very well.* »[f]

- The magazine *Electronics & Communications* announces Spectrix:
  The first major commercial computer operating system ever to be developed in Canada.
  The Spectrix operating system has a number of advantages over UNIX. It is real time. This should make it the operating system of the future for ... applications requiring guaranteed responsiveness.
  Spectrix also has a multi-mode operation, unlike UNIX.
  Spectrix is much more portable than UNIX.
  Much of the original research ... was performed at the University of Waterloo.

- *Maclean's* magazine writes:
  Despite its modest beginnings in 1967 as an appendage of the university's mathematics department, Waterloo's computer faculty is now producing one-quarter of the more than 1,350 computer science graduates in Canada. In a 1981 survey, Honeywell Inc. ... rated Waterloo first among the high-tech schools in all of North America, citing the university's advanced work in computer network technology ... (MIT) placed seventh.
  Not everyone is happy with Waterloo's approach ... Chairman of Dalhousie University's department of mathematics, computer science and statistics department, says that some colleagues are critical of the university's courting of corporations.
  ... Chairman of University of Toronto's Computer Systems Research Group ... argues that U of T is doing a better job in academic research, while Waterloo concentrates on applied software.
  If at times the campus begins to resemble an IBM branch plant, Brzozowski is not concerned.

«*Eric Manning recalls: "In the glory days (late 70s, early 80s) CS at Waterloo was freely compared with the best of the US schools and was often spoken of as one of the 10 best in N. America. Problems were in large part due to lack of $$; we were competing with US schools who had access to an entire funding system - the military, via DARPA, ONR, AFOSR, etc - which simply didn't and doesn't exist in Canada. Trying to compete with the BMWs on a VW budget caused enormous anxieties within the department."* »

- The Wall Street Journal writes:
  At Waterloo, more than a fifth of the 16,264 full-time students major

---

[f] Another chairman said to me: "Maybe you shouldn't touch on this delicate issue." (!?PJP)

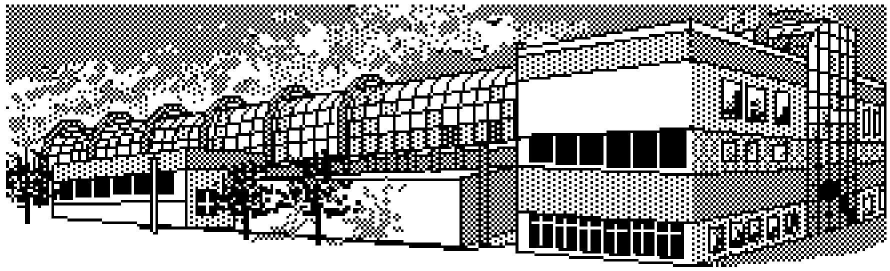
in math. Waterloo chases high-school math students the way Madison venue stalks soap consumers. "If you want to succeed, you have to be different," says Douglas Lawson. "If that involves acting a bit like an advertising firm, that's fine. It's worked for us."

- In 1982, the Institute for Computer Research (ICR) was established to strengthen industrial ties and enhance communication and collaboration between various (physically separated) groups on campus and Eric Manning appointed its first Director. The ICR was originally created as a loose federation of eight existing research groups with appointments in three academic departments spanning Mathematics and Engineering. That created some hard feelings: "If you didn't belong to a research group, you couldn't belong to the ICR". In 1984 the ICR was reorganized; individual researchers could now be associated with the ICR without belonging to a research group. Within a year of the reorganization the Institute had about 40 *individual* members as well as eleven "federated groups" (containing another thirty members): *Computer Communication Network Group, Computer Graphics Lab, Computer Systems Group, Data Encrypcion Group, Data Structuring Group, Logic Programming & Artificial Intelligence Group, Pattern Analysis & Machine Intelligence Group, Silicon Devices & Integrated Circuits Group, Software Portability Group, Symbolic Computation Group, VLSI Group.*

«*Eric Manning recalls "The idea of an ICR was due to Ian Blake of EE; he sold it to faculty. Put another way, he created my job as Director!"* »

By 1985 the Affiliates Program (wherein corporations receive access to research in return for fees) would encompass over 20 *principal affiliates* who receive invitations to seminars, general meetings, "presentation days" (at which graduate students describe their thesis projects) and research reports. In addition, the ICR hosted ten *Corporate Partners* who receive, in addition, office space in the Davis Centre for Computer Research (see below).

- In 1982 the Ontario government announced its plans to help build a home for the ICR: the Davis Centre for Computer Research (DC); it would contribute on a two-for-one basis, with UW providing $15.5 million. Completion of the DC was expected in 1986.



- Within a month of the above announcement, Digital Equipment Corporation (DEC), a founding *Corporate Partner* of the ICR, announced that it had entered into a $65 million research agreement with UW: Waterloo would provide $40 million in research personnel/space and DEC would provide $25 million in hardware, including about 15 VAX systems (11/780's and 785's) and about 2,000 personal computers (Rainbows and PRO 350's) and microVAX workstations.

«*This agreement was facilitated by a Federal government tax incentive to support research in Canada. I can recall "tax inspectors" visiting campus, in 1984, to see that the hardware was being used for research and not for teaching. Bev Marshman and I had received several PRO 350's for an Applied Math computer lab. Before the inspector arrived, we made sure the lab was judiciously labelled:*

MERLin - Mathematical Education Research Lab: innovare nexum . (PJP) »

- In 1982, Don Cowan, Wes Graham and Mike Malcolm warned the Ontario Ministry of Education of the folly of building a "Canadian Educational Microcomputer", (CEM) arguing that emphasis should be placed on portable software rather than quickly obsolescent hardware. They didn't listen. The following year a local high school, Bluevale Collegiate, was chosen as a test site for the all-new super-slick CEM micro: the $2500 Icon.

- In 1982, Ralph Stanton was welcomed back to Waterloo for a gala dinner. Ken Fryer had written a song, "Hello Ralphie":

Hello Ralphie, well hello Ralphie
It's so nice to have you back where you belong
You're looking swell Ralphie,
We can tell, Ralphie
You're still dreamin', you're still schemin',
You're still goin' strong.

You taught us well, Ralphie,
What the hell, Ralphie,

You're our favourite prof from way back when

So, fill up your glass, fellas,
Drink a toast to Ralph, fellas,
Wish he'd never go away
Wish he'd never go away
Wish he'd never go away again.

******

The close disciplinary association between Computer Science and Electrical Engineering lead naturally to close collaboration (CCNG, VLSI, the ICR, for example) - but it was not the only such collaboration. The Computer Graphics Group worked with the Dance Group in Fine Arts for entering and editing the Benesh notation for choreography, and with Psychology on colour perception and human factors in the design of graphics systems. But perhaps the most ambitious collaboration (between Computer Science, English and History) was the *New* Oxford English Dictionary project.

In 1984, the Oxford University Press (OUP) announced, simultaneously in London and in New York, that Waterloo would play a major role in computerizing the mammoth Oxford English Dictionary (OED).[g] Wes Graham spoke to the Royal Society:

"Waterloo will conduct a world-wide user survey ... our computer scientists will design a database ... our humanists will conduct research ..."

The President of the University of Waterloo, D.T. Wright (the prime-mover behind UW's involvement in the OED project) was delighted:

" ... it acknowledges our strength in the humanities ..."

******

In 1983, Bruce Simpson succeeded John Brzozowski as department Chair (although Doug Lawson filled in as interim Chair in 1983/84, while Bruce was on leave). Bruce recalls:

"I finished my PhD at U of Maryland (1966), and for several years after that I worked at two major US Engineering Schools and had

[g] The principal partners in the OED project were OUP, IBM-UK, ICC and UW. I can't believe that ICR in the DC and DEC weren't tangentially involved. See the OED section of this CSH. (PJP)

occasional 'touch-base' contact from Ralph Stanton[h] *(who I only knew by reputation). These Schools were typical of well established research institutes; substantial organizational and intellectual energy, but also substantial inertia associated with their status quo. It was difficult for such places to effect the changes necessary to accommodate the (then) new and ill defined area of Computer Science. Promoters of this amorphous prediscipline were typically fighting uphill, turf wars with existing academic departments which wanted to stake a claim on the potential of this glitzy new subject, but were suspicious of its intellectual substance, a suspicion largely born of ignorance. So, in 1971, when I came to UW for a year as a visitor to the CS department, it was a huge breath of fresh air to find CS with its own department, full of academics who had little doubt of the merit of what was congealing in Computer Science and on a level playing field with respect to other departments ... and with the wind at its back! I quickly decided to ask Don Cowan if I could stay."*

"I had a complete spectrum of experiences as chairman, from positive to negative, as probably each chairman has had. One of the most enjoyable events was developing the proposal for the ITRC - an opportunity for which Doug Wright deserves great credit. I initiated contact with Dereck Corneil - then chair of CS at Toronto, and also Jon Mark of EE here at Waterloo. Exploring cooperation with our former rivals was exciting, creative stuff, and the icing on the cake was being awarded the funding!"

In 1987, John Brzozowski returned once more to chair Computer Science.

After having spent three years as Director of the Division of Mathematics for Industry and Commerce (DMIC), Per-Åke G. (Paul) Larson became department Chair in 1989.

By 1992, the Computer Science Department had grown to some 17 Full Professors, 13 Associate Professors, 19 Assistant Professors, 3 Lecturers and 15 Adjunct Professors, with research groups which included:

Centre for the New Oxford English Dictionary, Text Research
Computer Graphics
Computer Networks and Distributed Systems
Computer Systems Group
Data Structures
Logic Programming and Artificial Intelligence

---

[h] Ralph often kept track of people ... *his* people.

Office Systems
Scientific Computing
Symbolic Computation : Maple
Theory of Computing
VLSI

Chapter Four

Symbolic Computation - Maple

Keith Geddes had called the meeting and posed the question.

******

Keith had become interested in "algebraic manipulation" while a graduate student at the University of Toronto (although a numerical analyst, part of his 1973 PhD thesis was on the symbolic computation of Pade approximations) and, while at Waterloo, had used a system available locally, ALTRAN:

- *very Fortran-like programming*
- *still batch processing*
- *restricted to polynomial and rational function manipulation.*

In the late 70's he began using MACSYMA "by dialing long-distance to the machine at MIT". At the time, MACSYMA only ran on a PDP-10 with lots of memory.

"The problem with MACSYMA was its resource-greedy implementation as a big Lisp application. You had to have a large *computer to run it on, and even then, you could not allow very many simultaneous users because it would completely bog down the system."*

In 1980 Keith was on sabbatical and had started a book on algebraic algorithms.[a] He recalls discussing with grad students at Berkeley the need for a computer algebra system which would fit on smaller machines (their general answer: no need to worry ... virtual memory is here!)

******

Now it was time to do something at UW.

In the Fall of 1980, he called the meeting of interested CS faculty.

Morven Gentleman, Gaston Gonnet[b] and Michael Malcolm were there to discuss Keith's question:

---

[a] ... and still not completed as of this writing! (PJP)

[b] The previous year Gaston had developed a little system called WAMA (Waterloo Algebraic Manipulation) to do calculations in asymptotic analysis and generally for analyzing algorithms and data structures.

"How are we going go to get , at Waterloo, a machine which is sufficient to run a system like MACSYMA?"

The response was clear:

- *forget the Lisp-based systems*
- *if we do it ,we should do it right*
- *start from scratch and use modern systems implementation languages (B on the Honeywell and later, C on Unix)*

By December, 1980 they had a little system running, the idea being to have a small compiled kernel and to let the bulk of the mathematical codes exist in an external library to be loaded only when needed. The external library code was executed by a Maple language interpreter, part of the compiled C kernel. *"A computer algebra system which could be used by many simultaneous users on a time-sharing system".*

Then they looked for a name.
Keith recalls:

"When I suggested the name Maple we decided to just leave it as a good Canadian name and not an acronym."

The small compiled kernel meant that Maple could also run on one-megabyte Macintoshes, Amigas, etc. By late in the 1980's, it would also run on DOS machines (once 32-bit 386s were introduced).

In 1982 the Toronto *Globe and Mail* announced: *"A Canadian computer language has been born at the University of Waterloo ... by substituting the fraction 1/3 instead of .333333 (and so on - stop us if you've heard it before) ... the mathematician can avoid gross miscalculation when dealing with zillions."*

******

Keith recalls developments within the Symbolic Computation Group:

• 1981
- *Greg Fee started working with us, then Bruce Char joined the department*

- *we moved from the Honeywell to a Vax 780*

- 1982
  - *the project was being funded by the UW Academic Development Fund*
  - *Benton Leong started working with us, then Stephen Watt and Michael Monagan as graduate students*

- 1983
  - *first major presentation, "The Design of Maple", at a conference (EUROCAL'83) in London, England*
  - *some skepticism about our design by which we would interpret the library code, rather than have compiled code*
  - *MAPLE was seen as a "toy system" by many people*
  - *we purchased our first "Unix box" for Maple, a Spectrix computer with a whole one megabyte of main memory. cost : $25,000*
- 1984
  - *WATDEC deal: major research collaborative agreement between Digital Equipment Corporation and UW, for the period 1984 - 1987*
  - *through WATDEC we obtained essentially all of our computing resources (mainly VAXes, later some microVAXes)*
  - *first major "NSERC Strategic Grant" funding for the project (to last for six years in total)*
  - *we made a presentation on "The Design and Performance of the Maple System" at the Macsyma Users' Conference*
  - *also at the first "Computers and Math" conference, held in New York City (here I first met Stephen Wolfram, with his SMP program)*
- 1985
  - *Maple Instructional Lab officially opened with 40 VT240 graphics terminals*
  - *first "commercial distribution" of Maple, through WATCOM*[c]
  - *now distributed on IBM VM/CMS, VAX VMS, and VAX UNIX*
  - *first "publication" of the Maple Users' Guide, through WATCOM*
- 1986
  - *I was local arrangements chairman for SYMSAC'86, here at the University of Waterloo*
  - *by now, this was a "logical choice" for the conference, indicating that we were "established" as a research group known to the research community (ACM SIGSAM)*
- 1987
  - *on sabbatical in Grenoble, France (working on the book again!)*
  - *concept of a commercial distribution company, Waterloo Maple Software, now being formed*
  - *in late 1987 / early 1988, Gaston and I had hired Michelle Dunkley to work on a business plan for the company*
  - *we also had Chris Kitowski, and Marta Gonnet, doing distribution*
  - *Brooks/Cole gearing up to distribute Macintosh Maple*

---

[c] These versions were sold for a one-year period after which they would self-destruct. I can recall being in Italy, listening to a computer scientist gleefully telling me they had defeated the self-destruct mechanism. Nevertheless, at the end of the year, their copy died. (PJP)

- 1988
  - *official incorporation of Waterloo Maple Software, April 1988*
  - *we operate a one-person company (then two-person) out of the "Community Small Business Centre" on Weber Street North*
  - *Mathematica is announced, June 1988 (big PR; good Macintosh interface; sexy 3-D graphics)*
- 1989
  - *in June, move Waterloo Maple Software to new premises on Columbia Street*
  - *new people: Janet Cater (Marketing); Stefan Vorkoetter (Technical Department); Joyce Brennan (Distribution)*
  - *became a six-person company during 1989*
  - *"acting CEO" is still Gaston and me*

- 1990
  - *good review in PC Magazine, May 1990*
  - *Maple V with X Window interface and 3-D graphics released*
  - *Ron Neumann joins the company as the full-time CEO*
  - *I visit distribution prospects in Japan, locating Mr. Iguchi of Cybernet Systems, who is with us here today*
- 1991
  - *Maple V*[d] *introduced, with "notebooks", 3-D graphics, the works*
  - *Maple has arrived on the commercial scene to stay*
- 1992 ∅ ∞ : what next?
  - *there's always more mathematical functionality*
  - *better user interfaces*
  - *interconnections between Maple and other programs (the "software bus")*

******

Steven Wolfram is the author of SMP, a symbolic computation program written while he was at Cal Tech. He moved to U of Illinois and was marketing the latest version: Mathematica. Yet Maple, its chief competitor, was making a name for itself in the world of symbolic computation.

Wolfram phoned Gaston Gonnet with threats to *"speak to your Dean"*.

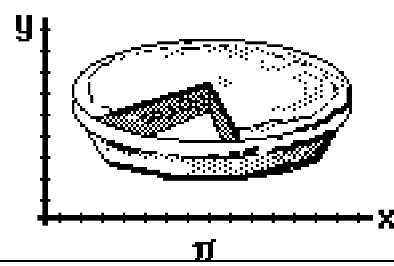
"Good," responded Gaston. *"I'll give you his name and number. One thing I should mention. Alan George is a member of the Maple Group."*

```
    |\^/|     Licensed for use at: University of Waterloo - MFCF
 _|\|   |/|_  WATCOM Products, Inc.
 \  MAPLE  /  Version 4.2
 <____ ____>  For on-line help, type  help();
      |


> y:=x^4*(1-x)^4/(1+x^2);
                                  4         4
```

---

[d] now requiring 2 megabytes of memory to support the graphics and other user-interface support.

```
                                x  (1 - x)
                          y := -------------
                                        2
                                   1 + x

 > int(y,x=0..1);
                          - Pi + 22/7
 > help(Pi);

 HELP FOR : initially-known names

 SYNOPSIS : The following names are known, either as global variables
             under user control or as names of constants known to certain
             functions.

  Catalan - Catalan's constant = sum( (-1)^i * 1/(2*i+1)^2, i=0..infinity );
            evalf(Catalan) is approximately 0.915965594...
  Digits  - number of digits carried in floats (default is 10)
  E       - base of natural logarithm;  evalf(E) is approx. 2.718281828...
  I       - complex number such that  I^2 = -1  (understood by "evalc")
  NULL    - initialized to the null expression sequence
  Order   - truncation order for taylor series (default is 6)
  Pi      - math constant pi;  evalf(Pi) is approximately 3.14159265...
  constants - See  help('constants');  ( default is the sequence:
                false, gamma, infinity, true, Catalan, E, I, Pi )
  false   - the value "false" in the context of Boolean evaluation
  gamma   - Euler's constant = limit( sum(1/i, i=1..n) - ln(n), n=infinity )
               evalf(gamma) is approximately 0.5772156649...
  infinity    - name for "complex infinity" used by some library functions
  lasterror   - See  help('lasterror');
  libname     - path name which is the root of the standard Maple library
  `mod`       - initially assigned the function name modp;  for symmetric
                  representation, assign `mod` := mods; (mod is an operator)
  prettyprint - switch (0 - off, 1 - centered, 2 - left justified) for
                   two-dimensional display of output (default is 1)
  printlevel  - See  help('printlevel');  (default is 1)
  prompt      - interactive prompt -- any string ( default is `> ` )
  screenheight - height of lines for character plots (default is 24)
  screenwidth - width of lines for printing (default is 79)
  status      - See  help('status');
  true        - the value "true" in the context of Boolean evaluation

 SEE ALSO : inifcns, `mod`, 'constants', 'lasterror', 'libname',
           'printlevel', 'status'

> plot(Pi);
```



Chapter Five

the *New* Oxford English Dictionary Project

"Just another database."
"Boring ... drudgery."
"Not likely a research problem."

Frank Tompa and Gaston Gonnet had agreed on every point. They weren't interested. The party was winding down; they drank another glass of wine.

" ... well, maybe there is *a research angle."*

******

E. Michael Brookes had been with UW in the early days and was now at Oxford University. When he learned that the Oxford University Press (OUP) would computerize the prestigious Oxford English Dictionary (OED) he immediately contacted Doug Wright, President of the University of Waterloo.

Doug flew to England.

The OUP was to put out a request for proposals to computerize the dictionary and it would be considered by more than a dozen organizations in the U.K. and North America[a]. Doug hoped to guarantee that Waterloo would be involved. When he returned from England he set up a group including Wes Graham, Don Cowan, Bruce Uttley and Jack Gray (an English professor), John Stubbs (a History professor who became co-director of the project), and Phil Smith (of the Arts Faculty computing facility). J. Minas, UW Computing Officer (and previous Dean of Arts), was to coordinate the University's response to the Oxford request.

It was a gargantuan project: the OED had been published in fascicles

[a] Waterloo was, in fact, the only non-commercial organization to respond to the request.

from 1884[b] to 1928 and eventually consolidated into a twelve-volume format with a one-volume supplement. The supplement grew to four-volumes between 1972 and 1986. The OED and Supplement comprised some 21000 pages, 66000 entries, 2 million quotations, 60 million words, over 350 million characters.[c] The word "set" alone occupied 21 pages, with 216 different meanings!

Oxford University Press had identified several independent phases:

1: <u>Publication of the integrated dictionary</u>
The text of the OED and its Supplement must be captured in machine-readable form, the Supplement integrated into the OED, and the integrated dictionary published by late 1989.

2: <u>Permanent database development for the *New* OED</u>
The machine-readable text for the integrated dictionary must be stored in a form suitable for subsequent maintenance of materials. In particular, a database and its attendant software must be designed to allow for future editing, revisions, additions, enhancements, publication of derivative dictionaries (national dictionaries,[d] subject-specific dictionaries) and provide facilities for processing complex queries.

It became clear that a great deal of technical expertise was required for Phase 2, and Frank Tompa and Gaston Gonnet were approached.

******

But Frank and Gaston weren't interested.
"Just another database."
"... not too interesting ..."
" Not really a computer science research problem."

The bottles of wine vanished, one-by-one, as the party progressed.
Then Gaston had a change of heart and approached Frank.

---

[b] The first volume, published Feb.1, 1884, covered *A* to *Ant*. It would be 44 years before the *Z*'s were completed!

[c] In 1970, the last historical entry for the word *computer* refers to computer typesetting developments, but with a quotation: "One of the unnerving things is the pace at which developments take place". Earlier meanings of *computer* referred to a person who computes, not a device.

[d] The Oxford University Press had approved an early version, modified in the USSR, which describes "Socialism" as a *social and economic system which is replacing capitalism. (As I write this, the USSR is no longer! (PJP))*

The OED was, after all, the only major historical dictionary of the English language, with a commitment to record the development of the language since c.1150. It was of vital importance to international scholars and researchers.

"It may not be important to computer science, but it's important to the world."

"Why not?"

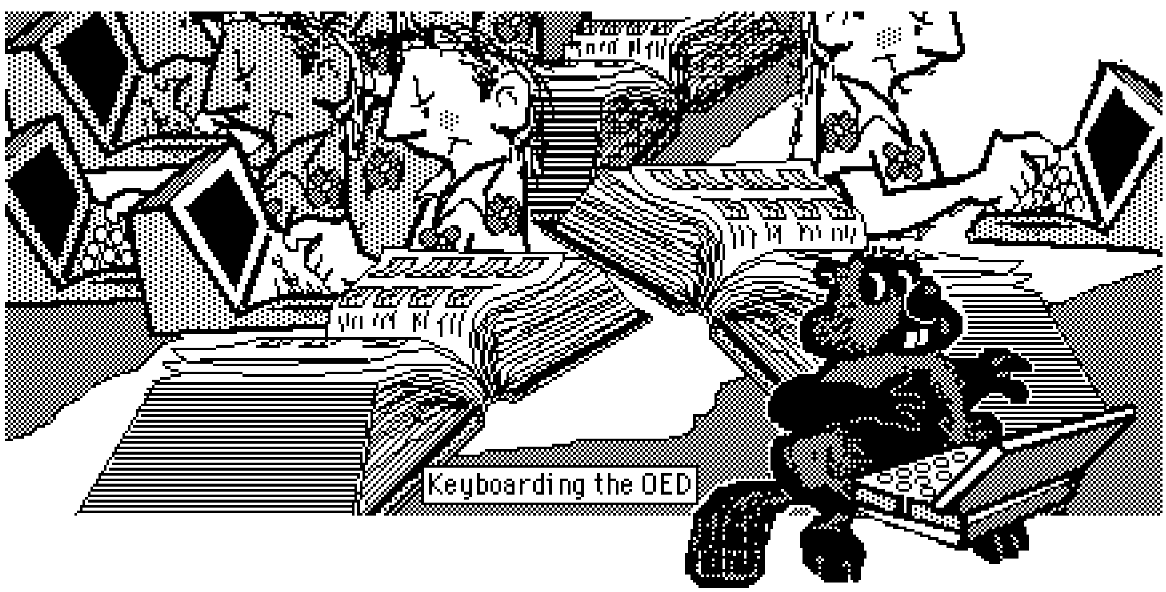
They agreed.

Gaston would contribute up to 50% of his time. (This was an upper bound since Gaston was also actively involved in the Maple project.)

Frank would contribute up to 30% of his time. (Little did he know how this would grow!)

Frank wrote up a proposal and conditions for their involvement, including reduced teaching load and a shoestring budget.[e]

When an OUP group toured North America, they stopped at Waterloo and listened to the UW proposal.[f] Subsequently, Tim Benbow (OUP project manager), Wes Graham, Peter Sprung (in the UW Dept. of Computing Services) and others, drew up a *Heads of Agreement* between the OUP and UW. Among other things it noted that the OUP would own the data while UW would own the software.



In May, 1984 (almost exactly one century after the publication of the

[e] Frank's "shoestring budget" was eventually doubled by Gaston, then by Wes, then by Wright. By 1989, approximately $2 million dollars would be spent on the project.

[f] The Oxford group was entertained at the Waterloo Motor Inn. Doug Wright, Wes Graham and Richard Charkin, a deputy publisher from the OUP, "really hit it off".

first OED volume!) the Oxford University Press made public its intention to computerize the dictionary. It had selected the participants:

- A U.S. firm, International Computaprint Corporation (ICC) would undertake manually keying-in the 350 million characters.[g]
- IBM (UK) donated £1 million in hardware, software and personnel for Phase 1.
- The University of Waterloo would assist in achieving Phase 2.

Wes Graham was to make a presentation before the prestigious Royal Society. Recognizing its historic importance, he spent weeks preparing a speech and sent a copy to Richard Charkin (((deputy publisher, OUP) and Charkin replied: "Your speech is awful. Scrap it. Just ad-lib for 5 minutes." Wes talked to Doug Wright. "He's kidding, isn't he? Just ad-lib ... to the Royal Society?" Doug agreed with Charkin - and Wes just "winged it" for five minutes.

The OUP anticipated the public response:

Q: *Why a U.S. firm to do the keyboarding?*
A: ICC is a subsidiary of the British firm, Reed International.

Q: *A Canadian University does the database? Was a British University approached?"*
A: *No* university was approached. We asked for responses, and Waterloo responded.

President Wright predicted that Waterloo involvement with the OED would last for more than 100 years.[h]

But there was some question as to whether Waterloo was up to the task:

"The question is whether they can design a database of such huge size in a creative way, so that you can get at the information easily," said a professor of linguistics at Brown University. *"I don't know whether they can do it - you need people who know both software and linguistics, and they are*

---

[g] Uneven print quality of the old metal plates and the complexity of fonts ruled out optical character readers.

[h] Doug Wright is reported to have said: "As President of UW I'd like to be remembered for two things: good food at the university club ... and the OED project."

hard to come by."

But Wright had put together the Waterloo group to cover all bases. The university-wide project included the Math Faculty and was centred in Arts "primarily because," as Jack Gray put it, "at its heart, the project deals with the analysis and manipulation of language".

In the Fall of 1984, UW received from ICC the first five pages of the letter "M", in machine-readable form.[i] Each entry contained the printed text: headword, pronunciation, part of speech, variant forms, historical derivations and roots, cross references to other parts of the OED and the variety of meanings (each of which contains a definition together with quotations illustrating the use of the word). The various components were often identified by changes in type styles and weight. Additionally, there were a number of structural "tags" which indicated where fonts shifted or where sections of an entry began. In the Supplement there would also be various *directives* such as "Add" or "Delete Now Obs."

It took several days to parse the first line!

It became clear that Phase 2 could only be accomplished if the data generated by Phase 1 were in an appropriate format - so UW became involved in Phase 1 as well. Tagging must be enhanced in order to support typesetting, integration of the Supplement with the OED, and database activities. A grammar was devised and a Waterloo computer science graduate student, Rick Kazman, spent eight months in Oxford working with the lexicographers to describe the required structure of the dictionary entries, using technology created by Howard Johnson, a Waterloo CS professor (as an outgrowth if his PhD research at UW). ICC tags were transformed to tags using Standard General Markup Language (SGML).

The OED entry for the word **Macco** begins:

Macco (mæ.ko). ?*Obs*. [? A variant spelling of MACAO.] A gambling game;

ICC provided the following:

Macco +PR (m+23 +11 k+I o+R). +LA +R ?+I +O Obs. +ET +O +SR A gambling game;

---

[i] Although the computer science researchers were delighted, there was little the Arts faculty could do, academically, with "5 pages of M".

With enhancements and SGML, it became:

```
<E><HG><HL>macco</HL><MPR>m&ae..k<i>o</i></MPR><IPR><IPH>&sm.m&ae
.k&schwa.&shtu.</IPH></IPR>.</HG>?<LB>Obs.</LB><ET>? A variant spelling of
<XR><XL>macao</XL></XR>.</ET><S4><S6><DEF> A gambling game;
```

... and the 350 million characters of the OED + Supplement blossomed to 550 million!

In January 1985, the University established the Centre for the New Oxford English Dictionary. A grant of $400,000 was awarded by the Secretary of State for Canada under a one-time "Centres of Specialization" program.[j] In 1986 the Centre received $50,000 from the Henry White Kinnear Foundation of Toronto and in November, 1986, the National Sciences and Engineering Research Council (NSERC) of Canada awarded $1.3 million over a three-year period to support research in "text-dominated databases".

By 1987/88 the project had shifted heavily toward the technical side and the original OED group was replaced by a research group entirely within computer science, with Frank Tompa and Gaston Gonnet as co-directors.

A data extraction language was developed (Goedel: derived from Waterloo's Maple system for algebraic computation) as well as a system to display the results (Lector) and Pat, a query system to answer questions:

How many words have opposite meanings?[k]

Which words were borrowed from the Italian during the seventeenth century?

When does the word "computer" first appear in the language?[l]

Early in the OED project, Frank and Gaston had visited the co-editor of the OED, in Oxford. While sitting in his office someone stuck his head through the door and muttered "they're tossing out the old OED plates ... for scrap." The editor paled, then they all jumped up and ran to the warehouse. Some 21,000 lead and copper plates were stacked from floor to ceiling,

---

[j] These "Centres" seemed an election ploy of the Liberal government of Canada. By the time the grant was presented, the Liberals had lost the election and it was a Conservative MP, Walter MacLean, who made the formal presentation.

[k] Like "cleave", meaning both *clinging together* and *splitting apart.*

[l] In 1897 a certain Mr. W. Cox describes a circular slide rule *computer*.

covering an entire wall. They would be melted down and the scrap metal sold. "Uh ... couldn't we take a few?" - and Gaston and Frank each took a few of the historic plates (making sure that the *first five pages of "M"* were among them). After successfully negotiating the airport X-ray machines, Frank returned to Waterloo with lead plates in hand and gave one to each of Doug Wright, John Stubbs and others involved in the project. He later *bought* one for his mother.[m] One hangs framed on his wall.

In 1987, Premier Bill Davis visited campus for the official opening of the *William G. Davis Centre for Computer Research.* He was given a demonstration of the OED software.

Searching for the number of occurrences of the word "Davis" : *1476*
Then, the number of occurrences of the word "Ontario" : *251*
Then, the number of instances where "Davis" occurs *near* "Ontario" : *2*

The first of these two OED entries was displayed:

ridership. [f. RIDER *sb.* + -SHIP.]
1. The position or office of a rider; rangership *(0bs.) rare.*
1450 *Rolls of Parlt. V. 192/1* The office of Ridership within the Forest of Delamare.**1485** *Rolls of Parlt. VI 382/1* The office of Rydership or Maister Forster of the Forest or Chase of Dertmore.
2. *orig. N. Amer.* The number of passengers (using a particular form of public transport). Also *attrib.*
1972 *W.G. Davis Urban Transportation for Ontario 5* This emphasis on the needs of the passenger and the improvement of service has enlarged the ridership considerably.

Bill Davis had contributed a new meaning to a very old word![n]

And the second entry?
Under the headword **premier** is the meaning:

c. *Austral.* and *Canad.* The chief minister of a State or Province.

And, as an example:

1972 *Ann. Reg. 1971* 79 On 21 October the Progressive Conservative Party of Ontario under a new Premier, Mr. William Davis, retained power with an increased majority winning 78 seats.

---

[m] Apparently, OUP eventually kept 1000 plates and sold them each for £ 30.

[n] As though to confirm that this new meaning had indeed become entrenched in the language, the OED goes on to note other, later uses of "ridership" (meaning *number of passengers on public transport*).

Bill Davis was delighted: "See who they select as an example of *premier!*"

******

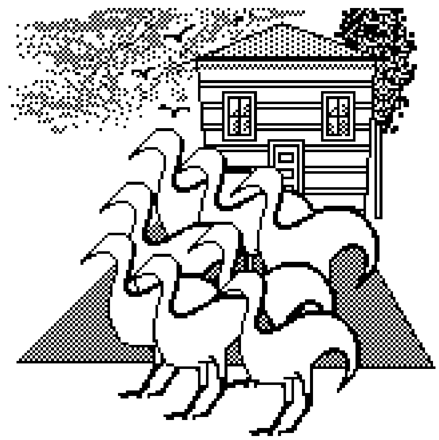
Eventually dozens of co-op students became involved in the OED project, a number of graduate students (for example there are 2 M.Math and 4 PhD students participating, in 1992), research programmers and faculty. "

Open Text Corporation" was established to commercialize the software, with UW and various faculty as shareholders.

The OED project had become, after all, a *valid computer science research problem* ... and it was <u>far</u> from boring!

Chapter Six

Computer Graphics



"You're invited to a party."

Kellogg S. Booth was speaking to some grad students in computer graphics.

"Oh, one other thing. He's a friend of mine and I happen to know that he likes pink flamingos."

The host didn't know the students were coming ... yet, the party was a big success.

The following morning a real estate agent showed up with a prospective buyer for the house - the house where the party had been held the night before.

There were 40 pink flamingos on the front lawn, courtesy of the students. For as long as they survived, the flamingos appeared at every graphics party for 5^{+} years. Understandably, the "pink flamingo" became the logo for the Graphics Group.

******

Kelly Booth and John Beatty had both joined the Computer Science department from the University of California at Berkeley by way of the Lawrence Livermore National Laboratory (Booth in 1977, Beatty in 1978). Both had *backed into* computer graphics, but by 1979 they had established a research group, acquired some equipment and started teaching a course in graphics. The next year they were awarded an NRC equipment grant and purchased an Ikonas frame buffer.

"This gave us a state-of-the-art raster display system on which much of our research over the next four years was performed. We later got a second Ikonas. Both are still used ... because nothing has ever matched their versatility."

Kelly Booth (now at the University of British Columbia) recalls:

"We grew a lot after that ... a well-equipped laboratory with a large

mainframe computer, specialized graphics processors, and a distributed network of workstations that continues to evolve. Each of the faculty members in the Graphics Group had a different background (computer science, physics, mathematics, ...). None of us did graduate work in computer graphics. Some of our research projects are hard-core computer graphics, but many are multidisciplinary. We work with faculty members in dance, chemistry, engineering, fine arts, and psychology. A mix of theory and practice existed in all of our work ... and both are important."

******

Richard Bartels had worked in numerical optimization, collaborating with Andy Conn (Andy had a foot in C&O and/or either/or CS ... depending upon the turmoil in the C&O dept.). Then Richard joined the CS department in 1979. (By then, Andy was 100% CS.)

But Richard's office was near John Beatty. They discussed/debated the notion of "splines" and found that numerical analysts and computer scientists had different understandings ... so they met on Saturdays to "teach each other", wrote a book on the subject and became instant "experts" in the graphics community.

In 1981, Richard joined the Graphics Group, the same year that the Computer Graphics Laboratory was formally instituted.

"CS was a jolly place in those days. Beatty and Booth shared a devil-may-care attitude ... John the organizational wizard, hassling the authorities ... Kelly flogging all the latest ideas. I just went along ..."

Richard chaired SIGNUM (1983-85), SIGGRAPH '83 was co-chaired by Beatty and Booth, and Kelly chaired SIGGRAPH (1985-87).

Kelly says:

"I don't see computer graphics as a having very many deep principles. Most of the principles come from fields outside of computer science (mathematics, physics, psychology, art, and philosophy, to name a few). The vast majority of people who actually do computer graphics for a living use it as a tool to solve problems in some domain other than computer science. Knowledge of the problem domain is at least as important as knowledge of the tool."

- From 1982 to 1989 there were just these three in the Group. Then Bill

Cowan[a] (who had collaborated with Kelly) left the National Research Council and joined the CS department.

- By 1992 the number of graduate students in graphics had grown from 5 to nearly 20.
- The number and variety of computing equipment grew (750, DEC 8600, two VAX 5400s, over twenty DEC workstations, several Adage RDS-3000 graphics systems, two SGI IRIS workstations, a SUN graphics workstation, nine experimental multi-processor workstations and a variety of graphical input-output devices[b]). The original PDP-11 is fondly remembered:

"Somehow John Beatty got it, second-hand ... a student-run Unix machine, it was one-of-two on campus and all hackers passed that way ... like Ian Allen, our technical guru." [c]

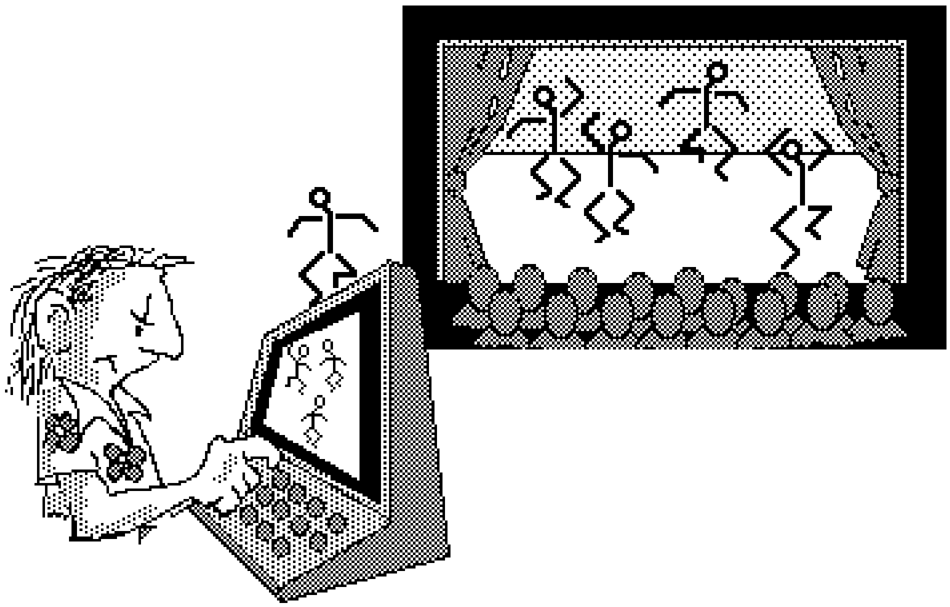
Research in the areas of *three-dimensional modelling of surfaces*, *human-computer interaction* and *the production of high quality digital colour images* included splines, synthesis of surfaces, ray-tracing algorithms, a message-passing language ("HARMONY", devised by Morven Gentleman), an early PAINT program (local artists were invited to use it, their work displayed in a subsequent "Art Show"), colour perception (jointly with Psychology) ... but perhaps few more intriguing than the Benesh editor (jointly with Dance Department in the Faculty of Applied Health Sciences).

---

[a] Bill is the son of Art Cowan, first chairman of Physics and a member of the "fearsome five" AAC. (See the *First Decade* of this CSH.)

[b] Huge gobs of computing power are required for computer graphics. An early graphics image required more computer time than was available on the Graphics Lab's machine, so a graduate student had two other MFCF machines working simultaneously one weekend. He was "caught" after some twenty four hours of computation (and image generation was terminated with eight scan lines to go!)

[c] See Appendix K: "Memories".



Appendix A

A memo to Arts Faculty Chairman : Oct. 25, 1965

Introductory Remarks Concerning the Faculty of Mathematics

One of the particular strengths of 20th Century science has been its willingness to modify situations when circumstances change. There was a day when the proton and electron were the only physical particles considered; now changing circumstances have made physicists recognize other particles. Similarly there was a day when it was quite appropriate for mathematics to be a single department within a faculty; changed circumstances now lead us to recommend a separate faculty. A number of items come to mind in this regard.

1. There is no doubt that a separate Faculty of Mathematics would save a great deal of administrative work. At the moment, we have to deal with the Arts Faculty, of which we are a part, and the Science Faculty, in which we give programmes. This leads to a great deal of extra work in running the Department. It would be much simpler if we controlled our own students, were able to recommend directly to Senate as regards admission and programmes, and handle other courses on a service basis. Since a Faculty is an administrative unit of convenient size rather than a mystical grouping of departments, it would seem natural to unite the more than 700 Mathematics students on campus in one Faculty. This number is growing rapidly and the complications of lack of autonomy increase markedly as our cooperative programme gains in enrolment and popularity.

2. As can be seen from the tentative calendar material, there is a basic unity in mathematical offerings. We shall be able to make our own admission and curricular requirements. It seems totally inappropriate to have a situation, as at present, where, in the determination of mathematical curricula in the Faculty of Science, the opinions of the Biology Department have a weight 3 times as great as those of the Mathematics Department. Such a situation is not logical.[a] Mathematical programmes should be set, administered, and controlled by mathematicians.

3. In my opinion, one of the most potent points is that Mathematics is neither Arts nor Science but stands in

---

[a] ... predating Mr. Spock.

between with characteristics of both. In fact, many of our difficulties stem from this fact. In the Arts Faculty, admission requirements are set with a view to other departments but not with a view to the needs of Mathematics. We are relegated to the position of "another subject" in Group C along with Religious Knowledge, Music, and Ukranian. On the other hand, in the Science Faculty, there is no concept on the part of some members of the special problems of mathematics students, and there have even been proposals that they should all be compelled to take a course in Biology.

4. Mathematics has developed in the past 10 years to include a host of new disciplines and these can best be developed in a separate faculty as Departments. For example, Statistics is a well differentiated discipline which is accepted as a separate Department in most universities. The same may be said of Computer Science. For disciplines such as these and others to develop, it is essential that they have a favourable milieu as would be provided in a Mathematics Faculty. So little is the recognition in some quarters of the vast developments of these areas that the Department of Biology has even proposed that freshman Mathematics students registered in the Faculty of Science (at present) should not be permitted to take our first year course in Computer Science. I believe that interference of this strange nature is bound to persist as long as Mathematics does not have appropriate autonomy.

5. I should stress that the proposed change is an administrative one not a curricular one. Indeed, the students will have a richer and more flexible possibility for choice of electives under the new programme. In addition, they will be united in one group rather than having their destinies scattered and divided as at present. Indeed, the present enormous developments in Statistics and Computer Science have lead these disciplines to become separate and diverging departments at many institutions. A Faculty of Mathematics would unite these allied areas into one grouping and thus maintain a greater coherence.

6. It is hoped that the accompanying material will provide material for discussion; I should certainly be glad to talk it over individually with any Department Chairman who might wish to contact me.

R. G. Stanton

ey

Appendix B

Undergraduate Courses in Computer Science
(1976/77)

CS 112 Introduction to Solving Business Problems by Computer
CS 113 Modelling and Simulation for Business Applications
CS 114 The Computing Process
CS 115 Introduction to File Processing
CS 116 Introduction to Computers
CS 117 Applications and Implications of Computers
CS 118 Introduction to Scientific Problem Solving by Computer
CS 140 Introduction to Mathematical Problem Solving by Computer
CS 150 Introduction to Computer Science - Characteristics of Computers
CS 180 Introduction to Computer Science - Programming and File Processing
CS 210 Introduction to Numerical Computing
CS 240 Principles of Programming Languages and Data Structures
CS 330 Computer Applications in Business
CS 331 Computer Applications in Business
CS 340 Machine, Data and Program Structures
CS 342 Implementation Schemes for Programming Structures
CS 360 Introduction to the Theory of Computing
CS 369 Digital Networks
CS 370 Introduction to Scientific Computation: Numerical Algebra
CS 371 Introduction to Scientific Computation: Numerical Approximations
CS 446 Scientific Applications Software
CS 450 Computer Architecture
CS 452 Real Time Applications of Minicomputers
CS 454 Principles of Operating Systems
CS 456[a] Data Communications
CS 462 Formal Languages
CS 464 Computability and Recursive Function Theory
CS 466 Algorithm Design and Analysis
CS 468 Program Verification
CS 472 Numerical Algebra
CS 474 Numerical Approximation
CS 476 Numerical Solution of Differential and Integral Equations
CS 478 Partial Differential Equations
CS 482 Business Systems Analysis
CS 484 Simulation
CS 498 Advanced Topics in Computer Science
CS 499 Readings in Computer Science

---

[a] It was suggested, in Math Faculty Council, that the calendar read: "CS four-five-six, pick-up-sticks"

Appendix C

the Computer Systems Group

In 1959, there were some 100 computers in all of Canada (used primarily for data processing and research) and the only available high-level language (still in its development stage) was FORTRAN. Demand for software to support courses in numerical methods, social sciences, engineering design, etc. yielded early application packages such as CPM, PERT and COGO ... all based on FORTRAN.

"To provide adequate software to support educational needs" became the driving force behind what was to become the "Computer Systems Group". Although never envisioned as a research-oriented organization, it nevertheless became a leading-edge contributor to state-of-the-art *software engineering.*

And it all began in 1964 ... when the IBM 7040 was installed at the University of Waterloo, to stand beside the 1620 ...

<u>The WATFOR Project</u>
The 7040, although having a much faster CPU than the 1620, required two minutes to compile even the simplest FORTRAN program (since the compiler was designed to rely on slow I/O devices for interim storage). All available compilers for large-scale computers had similar performance characteristics. Four third-year students wrote an "in-core, compile-and-go" compiler which eliminated any use of I/O devices as interim storage. They completed the project during the summer of 1965, and it was called WATFOR, and it was *very* successful. Student jobs which took an hour now took less than a minute (exclusive of printing time which was done off-line on another computer, the IBM 1401). But WATFOR also had diagnostic ability. Rather than producing pages of octal dumps when an error occurred, WATFOR provided a diagnostic attached to the original FORTRAN statement which was in error. This technique not only saved extensive consulting effort with students, but researchers soon realized that they could use WATFOR effectively during the development of their own programs.

Requests for the WATFOR compiler mounted and during 1965/66 it was distributed free to dozens of institutions. Of course there were the usual "bugs", and a programme of support was instituted involving travel to conferences, postage, long distance charges, tape copying, etc. so it was decided to charge a modest fee ($300) to offset the cost of distribution and support. Enhancement costs were supported by the University of Waterloo because the compiler was a strategically important system in support of the academic goals of the University.

Other Early Developments (1965-1972)

During this period many similar developments occurred, some under the auspices of the Computing Centre, and some under what had become the Mathematics Faculty. All these developments (eg. WATFOR/360, WATFIV, LISP 1.5, SPECTRE, SPECTRE/MAP, ASMG, WATBOL, TRACE, PDSQUISH and Cafeteria Style Computing) were either directly related to the support of specific courses, or for support of the operating environment in the Computing Centre. In each case they were distributed for a small fee, and soon there were literally thousands of copies of University of Waterloo software licensed throughout the world.

Cafeteria Style Computing

With the development of WATFOR and the subsequent increase in volume of jobs, attention shifted to improvement in the delivery system or "user interface". Originally, decks of punched cards held together with elastic bands were placed in boxes outside the computer room; the computer output was subsequently added to this package and returned to the student. Eventually, as many as 25,000 student jobs were processed daily and it became prohibitively expensive to collect input and return output to the students. At times there were as many as 20 part-time employees involved in this cumbersome process!

In 1967, a system was developed which permitted the students to submit their jobs directly into a card reader and to retrieve their output immediately at an adjacent printer. Students would "queue" to use the card reader, so the system became known as "Cafeteria Style Computing". Eventually, such job submission facilities were duplicated at several points on the campus to provide remote access to the computing facilities.

Assembler G

In the 1960's, almost all systems development was accomplished using assembler language programming so it was important to have an efficient assembly language processor. Assembler G (ASMG) was developed, and was so fast it became the standard processor in general use including several IBM Research Labs and Centres. (For example, IBM Houston used ASMG to develop HASP, a spooling package which revolutionized the use of batch operating systems).

WATFIV

In 1964, IBM announced the 360 computing system. It became the system of choice for most academic institutions, including Waterloo. It was therefore important to prepare a version of WATFOR for the 360. This vastly-improved implementation was eventually called WATFIV (1967), and became the most widely-used of the various WATFOR developments. WATFIV continues to be popular; over two hundred installations still run this software. WATFIV and its authors were the recipients of a number of awards. In 1972, the project leaders were awarded the Grace Murray Hopper Award by the ACM to recognize their accomplishment, and in 1975, 1976 and 1979 WATFIV received the Datapro Software Honour Roll and Award of Merit.

WATBOL

With the introduction of the co-operative education programme in Computer Science (mid 1960's), it became important to use COBOL as the language for implementation of data processing algorithms. WATBOL was developed and had operating characteristics similar to WATFOR. (It is still used in hundreds of educational institutions.)

Artificial Intelligence

By 1966 LISP had been recognized as making an important contribution to Artificial Intelligence. No accurate implementation on the IBM 360 was available, so the University of Waterloo created its own version of LISP 1.5. It was distributed to about 50 universities, including Stanford. Stanford subsequently incorporated the University of Waterloo code into their system and further distributed it. Subsequently a software implementation of PROLOG was developed (1977) which is referenced in the literature as a "pioneering effort". Much later, CSG ported a UNIX version of Waterloo PROLOG to the IBM mainframe computers.

Formalizing the COMPUTER SYSTEMS GROUP (1972-1980)

In 1972 the Computer Systems Group (CSG) was officially approved by the University of Waterloo Board of Governors. One result of this formal approval was to provide a constitution, by-laws and a Board of Directors which formally defined the status and future goals and activities of CSG. Further, CSG was given the mandate to develop, distribute, maintain and support its own new systems. As well, it would continue to distribute and support existing systems in addition to new ones created by others members of the University community. It was to be a financially independent operation, with all employees on a "soft money" basis. CSG had, in many respects, become a software development business within the University. There was, however, an important constraint; the systems developed by CSG must support the academic and administrative activities of the University. Because of this campus-wide mandate, the first Chairman of the Board was the University Treasurer, who, as the chief financial officer, had the entire University's interests in perspective. Further, the CSG Board consisted of representatives throughout the University community. With its broadened mandate, CSG began to grow rapidly, new systems were developed and used effectively to support the academic process. By 1980, the world-wide success of these systems resulted in income exceeding $1,000,000 per year and CSG was able to make substantial contributions in equipment for course work supported by the Computing Centre. Further, donations for scholarships in mathematics were made on an annual basis to the Rene Descartes Foundation and CSG employees became involved in the teaching of many undergraduate and graduate courses, particularly in the Math Faculty.

WIDJET

By 1974 Cafeteria Style computing had become cumbersome. Long line-ups formed at each card reader, the campus was becoming "paved" with key punches and it was estimated that the University of Waterloo was spending $300,000 per year on cards and paper to handle student jobs. The Chairman of the Computer Science Department approached CSG to develop a job-submission system which would reduce some of the costs of consumables and introduce students to state-of-the-art data-input technology. In this new concept students would use a mini-computer-based multi-terminal system for editing programs. The programs would then be submitted over a high-speed communication link to the mainframe for processing. Results were routed back to the mini and displayed using the editor. CSG approached Digital Equipment Corporation (DEC) for advice because it was felt that the PDP-11 was the appropriate minicomputer to use. DEC officials were skeptical; the terminal interface hardware could not handle the volume. Yet, CSG experiments indicated positive response times so the project was continued and in September 1975 the first of many WIDJET systems was placed into productive use. WIDJET virtually eliminated punched cards on the UW campus, and was the first (and probably the only) system of its kind. It was well received at other educational institutions, and its use at Waterloo was continued until 1984 (when it was replaced by local area networks of microcomputers). In 1977, WIDJET was also made available on the IBM Series/1. Both DEC and IBM were pleased and pursued co-operative marketing arrangements for the system.

WATFIV/S
In 1973, it was important to provide Waterloo students with access to "structured programming" so CSG extended the FORTRAN language to include the WHILE and IF-THEN-ELSE control constructs. WATFIV was then modified and became known as "structured WATFIV" or WATFIV/S. At the same time the specification of character variables proposed by a user group was implemented in the compiler. (These developments had a significant impact on the deliberations of the FORTRAN standards committee, and subsequently on the FORTRAN-77 standard.)

Waterloo Pascal
For the purpose of teaching the principles of programming to second-year students, Pascal became the language of choice. There were early Pascal compilers available on the mainframe but they were inappropriate for teaching, tending to be unreliable and/or slow. Thus CSG, at the request of the Computer Science Department, prepared the Waterloo Pascal processor with functional characteristics similar to WATFOR.

WATIAC and WATMAP
In 1973, the Computer Science Department formed a committee to design a pseudo computer which could be emulated, and which would be useful for teaching hardware architecture and assembly language programming, including the concepts of linking, loading, binding, and macro processing. The resulting design specification was implemented by CSG and called WATIAC and WATMAP. The system was used until 1982 when it was replaced by the MicroWAT.

Waterloo SCRIPT
During the 1970's, SCRIPT became popular as an early form of batch-oriented desk-top publishing. A SCRIPT processor was originally developed by MIT, but in 1974 Waterloo's Computing Centre (DCS) agreed to assume responsibility for its future development. CSG distributed and provided first-level support for each successive version. For most computer users at the University of Waterloo, Waterloo SCRIPT was the only reasonable way to produce documents and the Waterloo SCRIPT processor eventually

supported the most commonly available typesetters and laser printers. The Waterloo SCRIPT processor was so well accepted off-campus that its popularity rivaled (and possibly exceeded) WATFOR. In 1979, Waterloo SCRIPT received the Datapro Award of Merit.

Microcomputing, Networking, etc. (1980-1992)

In 1979, with the encouragement of the Academic Vice-President, CSG began to work extensively with microcomputing technology. At that time no reasonable system was available on the market. (The Apple, TRS-80 and Commodore Pet had limited size and function and the IBM PC was not generally available until 1982).

The MicroWAT

During 1980, the MicroWAT was developed as a bus-oriented, open-architecture 6809 system with up to a megabyte of memory and I/O suitable for networking. Portability technology was used to develop the MicroWAT software which included language interpreters for BASIC, Pascal, FORTRAN, COBOL and APL. The MicroWAT circuitry was incorporated into the Commodore SUPERPET and the NABU personal computer, and was purchased by the IBM research labs in Poughkeepsie, New York. As well, CSG incorporated the MicroWAT circuitry into Volker Craig "dumb" terminals, thus converting them into personal computers. The design effort associated with the MicroWAT resulted in a donation to the University of 35 Commodore 8032 microcomputers which were subsequently modified to become Commodore SuperPets. Such systems were affordable and functional, and so were used to support many courses at the University of Waterloo (until the IBM PC arrived on the scene).

IBM and Waterloo PC Network

All of this activity came to the attention of IBM (United States) and resulted in the University of Waterloo being provided with IBM PC's, mainframes and minicomputers to create a campus-wide network called Waterloo PC Network. Total value: $4 million. This network became an IBM product, marketed throughout North America. Later, IBM (Canada) provided another $3.5 million in equipment, much of it to enhance this existing network.

JANET

In 1982, the Department of Computing Services (DCS) pioneered the development of the first version of the JANET[a] local area network to support a lab of IBM PC's in the Physics Department. In 1984, IBM introduced new local area network hardware for PC's, and as a consequence CSG developed a new version of the JANET system. IBM supported this research effort by donating 35 IBM PC's for JANET labs in Earth Sciences, and Biology. JANET became recognized in many countries, and as a result of its exposure in the Far East, IBM Japan commissioned CSG to create a Kanji version. By 1989 there would be 400 JANET networks installed throughout the world.

MacJANET

Several months prior to the MAC/LISA announcement, Apple provided CSG with prototype systems. By 1985, CSG was working extensively with the Apple Macintosh computer. The University had a specific goal: to modify a first-year service course, CS 100. The course material was best supported using a Macintosh, but a network was needed to reduce the cost and permit the sharing of data, software and printers. CSG developed MacJANET and Apple donated the 48 computers required to offer the course. In 1988, CSG modified MacJANET to have Ethernet and multi-server support so it could be used for all first-year core courses in Computer Science. Hundreds of MacJANET workstations have since been installed on campus and the network is used by hundreds of other educational institutions, mainly in North America, and is co-operatively marketed by Apple Canada.

Portable Computers - Project Aries

In 1984, CSG became involved with portable computers in Project ARIES (Applied Research in Educational Systems). This resulted in the gift of hundreds of portable computers from several manufacturers, as well as the establishment of a campus-wide network to provide access to software, data and I/O devices. The project was also supported by government grants amounting to more than $1.4 million. Technical development and support were undertaken by CSG with educational research being done by academics in all faculties, for both on-campus and distance education students.

PWAP

In 1985, at the request of the Computer Science Department, CSG converted one of its parser generators (PWAP) for use in the course CS 240. PWAP

---

[a] JANET stands for **J**ust **A**nother **NET**work, a name suggested by Paul Dirksen. Because it originally involved communication over an IEEE-488 port (on the Commodore PET), the first name suggested was ICON, for **I**EEE **CO**mputer **N**etwork. Since the two major contributors to the JANET design were Jerry Bolce and Adrian Weerheim, it was also called the **J**erry and **A**drian **NET**work. See Appendix K.

would assist the students in writing compilers by allowing them to produce table-driven parsers in Pascal. In 1988, CSG ported PWAP to the Macintosh for use in the new MacJANET lab for courses CS 131 and CS 132.

Modula-2

In 1986, CSG developed a Modula-2 compiler with the same functional characteristics as earlier educational compilers (such as WATFOR and Waterloo Pascal). The Modula-2 language is a Pascal derivative which has strong data typing and provides other facilities for the development of much larger programs.

CSG and Portability

From the time programs were first developed for computers, portability was an issue. By 1980 the C language, C library and run-time system had become recognized as the standard for developing portable systems. However, for proprietary reasons, most software development laboratories felt they had to design their own C-like system. CSG was no exception; it developed WSL (Waterloo Systems Language, pronounced *whistle*). Although WSL served the purpose, C as a language soon came into the public domain, so CSG developed Waterloo C for IBM mainframe systems (regarded by many as the best available implementation for the IBM 370).

In 1982, IBM approached CSG to co-operate with their laboratories in Boca Raton to do an extensive portability exercise for a 68000-based system. As a result some CSG employees took up residence in Boca in the spring of 1983, and during a period of only 4 months successfully ported all of the Waterloo micro-based software. Needless to say IBM was suitably impressed, and this led to a further strengthening of relationships with the IBM labs.

In 1983, Hewlett-Packard approached the Computer Science department to do a joint study related to the use of its new series of 68000-based computers. Since there was insufficient software, CSG agreed to port an editor and FORTRAN processor so the rest of the study could be conducted. This was successful, and a small local area network of these high-performance workstations was donated to the CS department.

User Interfaces

In 1987, CSG was involved in the creation of a User Interface to provide a consistent look and feel to all CSG software. CSG also developed a menu-driven Application Shell which could be used as an alternative to traditional command-oriented operating systems interfaces.

CSG has recently become involved in developing better tools to deal with graphical user interfaces in a portable manner. One such tool is a user interface generator which will allow the user interface portion of an application to be written so it is portable; the appropriate code will be generated for various manufacturers' user interfaces.

CSG is also experimenting with porting the current user interfaces to a UNIX environment using CURSES (a terminal interface available on all UNIX systems). IBM Canada has become interested in these activities, providing over $950,000 worth of equipment and software.

Databases

In 1988, CSG became involved in databases primarily from the user perspective. There has been substantial experimentation with structured data, and some preliminary work on databases for use in multi-media. Particular attention has been paid to user interfaces and to multiple databases which are distributed and connected by communication networks. The research in databases is now focussing on database tools designed for non-technical users. The goal is to create portable software tools which produce SQL, so that the user may interact with many different implementations of SQL databases such as ORACLE, SQL/DS, and DB/2.

CSG has recently worked on a research contract for Bell Canada on user interfaces and databases. The specific research examined the X.500 Directory Standard, one of the many components of the OSI model. This work resulted in a $112,000 contract and an ensuing $112,000 research grant.

Epilogue

At the March, 1992 meeting of Math Faculty Council, Dean Jack Kalbfleisch reported:

"The Computer Systems Group (CSG) is one of the oldest research groups in the University. It was formed in 1968 by an act of the Board of Governors and has continued to be responsible to the Vice-President, Academic and the Board since then. At the April meeting of the Board, CSG was dissolved as a research group reporting to the Board and is now a research group within the Faculty of Mathematics. I am very pleased with this development; the Faculty of Mathematics has profited in the past through its association with

CSG, and will, I am sure, continue to do so. Eric Mackie has been Director of CSG and will continue to be closely involved along with Wes Graham. The new Director of CSG is Don Cowan from the Department of Computer Science."

Appendix D

PhD Theses in the Department of Computer Science

1991

Vinay Deshpande
Specification and Parallel Implementation of Joins for Flat and Nested Relations
(Supervisor: P.-A. Larson)

Bart C. Domzy
Performance of Network Protocols for Time-Constrained Applications
(Supervisor: J.W. Wong)

Vladimir Estivill-Castro
Sorting and Measures of Disorder
(Supervisor: D. Wood)

Daniel A. Ford
Performance Optimizations for Optical Disc Architectures
(Supervisor: S. Christodoulakis)

James W.K. Hong
Communication Abstractions for Distributed Systems
(Supervisor: J.P. Black)

Richard T. Hurley
An Investigation of File Migration in a Distributed File System
(Supervisor: J.W. Wong)

Peter Triantafillou
Employing Replication To Achieve High Availability and Efficiency In Distributed Systems
(Supervisor: D. Taylor)

Fei Song
A Processing Model for Temporal Analysis and its Application to Plan Recognition
(Supervisor: R. Cohen)

Bruce Spencer
Assimilation in Plan Recognition via Truth Maintenance with Reduced Redundancy
(Supervisor: R. Cohen)

Peter Van Beek
Exact and Approximate Reasoning about Temporal Relations
(Supervisor: R. Cohen)

1990

Eng-Wee Chionh
Base Points, Resultants, and the Implicit Representation of Rational Surfaces
(Supervisor: R.N. Goldman)

Bruce Cockburn
Fault Models and Tests for Random-Access Memories
(Supervisor: J.A. Brzozowski)

David Forsey
Motion Control and Surface Modeling of Articulated Figures in Computer Animation
(Supervisor: R. Bartels/K. Booth)

Pierre Gaboury
VLSI Architecture Design Using Predicate Logic
(Supervisor: M. Elmasry/D. Poole)

Anthony Joseph Gahlinger
Coherence and Satisfiability of Waveform Timing Specifications
(Supervisor: C.J. Colbourn)

Tony Lai
Efficient Maintenance of Binary Search Trees
(Supervisor: D. Wood)

Ronald McFadyen
Sequential Access in Files used for Partial Match Retrieval
(Supervisor: P.A. Larson)

Michael Burnett Monagan
Signatures + Abstract Types = Computer Algebra - Intermediate Expression Swell
(Supervisor: G. Gonnet)

Sanjay R. Radia
Names, Contexts, and Closure Mechanisms in Distributed Computing Environments
(Supervisor: J. Pachl)

Daniel Joseph Salomon
Scannerless Parsing of Programming Languages
(Supervisor: G.V. Cormack)

Trevor Janathan Smedley
Fast Methods for Computation with Algebraic Numbers
(Supervisor: K.O. Geddes)

J. Andre N. Trudel
Representing and Reasoning about a Dynamic World
(Supervisor: R. Goebel)

Paul van Arragon
User-Modeling by Nested Default Reasoning
(Supervisor: D. Poole)

1989

Hosam M.F. AboElFotoh,
Reliability of Radio Broadcast Networks: A Graph Theoretic Approach
(Supervisor: C.J. Colbourn)

Ricardo A. Baeza-Yates,
Efficient Text Searching
(Supervisor: G.H. Gonnet)

Wing Hong Cheung,
Process and Event Abstraction for Debugging Distributed Programs
(Supervisor: J.P. Black, E.G. Manning)

Ian J. Davis,
Error Correction in Robust Storage Structures
(Supervisor: D.J. Taylor)

Eduardo Francisco D'Azevedo,
On Optimal Triangulation for Piecewise Linear Approximation
(Supervisor: R.B. Simpson)

Robert Victor Klassen
Device Dependent Image Construction
(Supervisor: W.B. Cowan, K.S. Booth)

Louis D. Nel,
Network Reliability and Facility Location in Unreliable Networks
(Supervisor: C.J. Colbourn)

Eric Neufeld,
Construction of a Sound Nonnumeric Probabilistic Reasoner
(Supervisors: R. Aleliunas, D. Poole)

Dimpy Pathria
The Numerical Analysis of Certain Nonlinear Dispersive Wave Equations
(Supervisor: J. Ll. Morris)

1988

Eleanor C. H. Chu,
Orthogonal Decomposition of Dense and Sparse Matrices on Multiprocessors
(Supervisor: J.A. George)

Neil Coburn,
Updating Derived Relations
(Supervisor: P.-A. Larson)

H. Douglas Dykeman,
Architecture and Performance of Information Delivery Systems
(Supervisor: J.W. Wong)

Eric S. Fraga,
Adaptive Mesh Refinement Techniques for Nonlinear Dispersive Wave Equations
(Supervisor: J. Ll. Morris)

Rangaswamy Jagannathan,
A Descriptive Prescriptive Model for Dataflow Semantics
(Supervisor: E.A. Ashcroft)

Yuying Li,
An Efficient Algorithm for Nonlinear Minimax Problems
(Supervisor: A.R. Conn)

Wendy J. Myrvold,
The Ally and Adversary Reconstruction Problems
(Supervisor: C.J. Colbourn)
Carl-Johan Henry Seger,
Models and Algorithms for Race Analysis in Asynchronous Circuits
(Supervisor: J.A. Brzozowski)

Paul C. van Oorschot,
Combinatorial and Computational Issues Related to Finding Roots of Polynomials over Finite Fields
(Supervisor: S.A. Vanstone)

1987

Jose A. Blakeley,
Updating Materialized Database Views
(Supervisor: P.-A. Larson & F.W. Tompa)

Mantis Hoi Ming Cheng,
Lambda-Equational Logic Programming
(Supervisor: M.H. van Emden)

Gabriel de Vos Smit,
A Formatter-Independent Structured Document Preparation System
(Supervisor: D.D. Cowan)

Ehab S. El-Mallah,
Decomposition and Embedding Problems for Restricted Networks
(Supervisor: C.J. Colbourn)

Jose I. Icaza,
Adaptive Selection of Query Processing Strategies
(Supervisor: F.W. Tompa)

Gerald W. Neufeld,
A Name Service for Widely Distributed Computing Environments
(Supervisor: M.A. Malcolm)

Patrick A. D. Powell,
Icewater: A Procedural Design Language for VLSI
(Supervisor: M.I. Elmasry)

Aparna Ramanathan,
Improving Bounds for All-Terminal Network Reliability
(Supervisor: C.J. Colbourn)

Gregory J. E. Rawlins,
Explorations in Restricted Orientation Geometry
(Supervisor: D. Wood)

Rangaswamy Vasudevan,
Network Transparency in Multi-Process Structuring
(Supervisor: M.A. Malcolm)

Hong Zhi Yang,
Query Transformation
(Supervisor: P.-A. Larson)

Keitaro Yukawa,
Amalgamating Functional and

Relational Programming through the Use of Equality Axioms
(Supervisor: M.H. van Emden)

1986

Pedro Celis,
Robin Hood Hashing
(Supervisors: P.-A. Larson & J.I. Munro)

Joseph C. Culberson,
The Effect of Asymmetric Deletions on Binary Research Trees
(Supervisor: J.I. Munro)

Francis Chi-Moon Lau,
Policies and Mechanism for Distributed Clusters
(Supervisors: E.G. Manning & J.P. Black)

Medahalli V. Ramakrishna,
Perfect Hashing for External Files
(Supervisor: P.-A. Larson)

Jonathan Schaeffer,
Experiments in Search and Knowledge
(Supervisors: M.H. van Emden & R.G. Goebel)

Stephen M. Watt,
Bounded Parallelism in Computer Algebra
(Supervisor: K.O. Geddes)

Sheng Yu,
On Systolic Automata
(Supervisor: K. Culik)

1985

Richard J. Beach,
Setting Tables and Illustrations with Style
(Supervisor: K.S. Booth)

Senad Busovaca,
Handling Degeneracy in a Nonlinear L1 Algorithm
(Supervisor: R.H. Bartels)

Gita Gopal,
Reliable Broadcast Protocols for Local Area Networks
(Supervisor: J.W. Wong)

Prabandham Madan Gopal,
Voice Transmission on Local Area Networks
(Supervisor: J.W. Wong)

Rolf Karlsson,
Algorithms in a Restricted Universe
(Supervisor: J.I. Munro)

Claudia M. B. Medeiros,
A Validation Tool for Designing Database Views That Permit Updates
(Supervisor: F.W. Tompa)

1984

Barry Joe,
Finite Element Triangulation of Complex Regions Using Computational Geometry
(Supervisor: R.B. Simpson)

Howard Johnson,
Formal Models for String Similarity
(Supervisor: F.W. Tompa)

Walter Cunto Pucci,
Average Case Selection
(Supervisor: J.I. Munro)

Hamza Rashwan,
Auxiliary Storage Methods for Sparse Positive Definite Linear System
(Supervisor: J.A. George)

Paul Stachour,
Portability Verification of PL/1

Programs: Definition and Detection of Compiler Incompatibilities
(Supervisor: W.M. Gentleman)

Hideyuki Tokuda,
Shoshin: A Distributed Software Testbed
(Supervisor: E.G. Manning)

Ali Yazici,
An Organization of the Extrapolation Method of Multidimensional Quadrature for Vector and Multiprocessing
(Supervisor: R.B. Simpson)

1983

Janice I. Glasgow,
An Operational Semantics for Lucid and Its Correctness
(Supervisor: E.A. Ashcroft)

Esmond G. -Y. Ng,
Row Elimination in Sparse Matrices Using Rotations
(Supervisor: J.A. George)

Patricio Poblete,
Fringe Techniques for Binary Search Trees
(Supervisor: J.I. Munro)

Son T. Vuong,
Formal Techniques for Protocol Specification and Validation - Toward an Integrated System
(Supervisor: D.D. Cowan)

1982

James P. Black,
Analysis and Design of Systems of Robust Storage Structures
(Supervisors: D.J. Taylor & D.E. Morgan)

James H. Bradford,
The Eta Interface - An Error Correcting Parser for Augmented Transition Networks
(Supervisors: M.H. van Emden & T. Pietrzykowski)

Jan K. Pachl,
Formal Description of Communication Patterns
(Supervisor: K. Culik)

Alfredo R. Piquer,
A Reliable Replicated Storage Service for a Local Network
(Supervisor: M.A. Malcolm)

Nivio Ziviani,
The Fringe Analysis of Search Trees
(Supervisor: G.H. Gonnet)

1981

Erich Bretholz,
Optimization Techniques in Physical Data Base Design
(Supervisors: J.W. Graham & D. Rotem)

Paulo Roberto Freire Cunha,
Design and Analysis of Message Oriented Programs
(Supervisor: T.S.E. Maibaum)

Larry Rendell,
An Adaptive Plan for State-Space Problems
(Supervisor: T. Pietrzykowski)

1980

Thomas Todd Carey,
Reachability in Restricted Classes of Petri Nets
(Supervisor: E.A. Ashcroft)

Thomas Alan Cargill,
View of Source Text for Diversely Configurable Software
(Supervisor: W.M. Gentleman)

Norman John Livesey,
Run-Time Control in a Transaction-Oriented Operating System
(Supervisor: E.G. Manning)

John Kent Peacock,
Distributed Simulation Using a Network of Processors
(Supervisors: E.G. Manning & J.W. Wong)

Raul Javier Ramirez Inurrigarro,
Efficient Algorithms for Selecting Efficient Data Storage Structures
(Supervisor: F.W. Tompa)

Nicola Santoro,
Efficient Abstract Implementations for Relational Data Structures
(Supervisor: F.W. Tompa)

Hendra Suwanda,
Implicit Data Structures for the Dictionary Problem
(Supervisor: J.I. Munro)

Denis Thérien,
Classification of Regular Languages by Congruences
(Supervisor: J.A. Brzozowski)

Mehmet Sabri Unsoy,
Congestion Control in Hierarchical Networks
(Supervisors: J.W. Wong & D.E. Morgan)

1979

Monisoye Olorunsola Afolabi,
Symbolic Series Solution of Ordinary Differential Equations
(Supervisor: K.O. Geddes)

Carlos A. P. Carvalho,
On the Analysis of Programs with Equations and Binary Relations
(Supervisor: M.H. van Emden)

1978

David Ross Cheriton,
Multi-Process Structuring and the Thoth Operating System
(Supervisor: M.A. Malcolm)

Michael R. Levy,
Data Types with Sharing and Circularity
(Supervisor: T.S.E. Maibaum)

Tok-Wang Ling,
Improving Data Base Integrity on Functional Dependencies
(Supervisor: F.W. Tompa)

David Russell McIntyre,
Linear Time and Space Ordering Algorithms for Irregular Shaped Finite Element Systems
(Supervisor: J.A. George)

Sylvia L. Osborn,
Normal Forms for Relational Data Bases
(Supervisors: F.W. Tompa & J.I. Munro)

1977

Brian R. Allen,
On Binary Search Trees
(Supervisor: J.I. Munro)

Philip T. Cox,
Deduction Plans: A Graphical Proof Procedure for the First-Order Predicate Calculus
(Supervisor: T. Pietrzykowski)

Mansour Farah,
Correct Compilation of a Useful Subset of Lucid
(Supervisor: E.A. Ashcroft)

Gaston Gonnet,
Interpolation and Interpolation Hash

Searching
(Supervisor: J.A. George)

Mohamed G. Gouda,
Protocol Machines: Towards a Logical Theory of Communications and Communication Protocols
(Supervisor: E.G. Manning)

Matthew C. B. Hennessy,
A Formal Approach to the Study of Parameter Passing Mechanisms and Non-Determinism
(Supervisor: E.A. Ashcroft)

Marek I. Irland,
Analysis and Simulation of Congestion in Packet-Switched Networks
(Supervisor: E.G. Manning)

Richard C. Kolanko,
A Structured Approach to Performance Measurement of Computer Networks
(Supervisor: D.E. Morgan)

David W. H. Szeto,
A Functional Model for Data Bases
(Supervisor: J.G. Linders)

David J. Taylor,
Robust Data Structure Implementations for Software Reliability,
(Supervisor: D.E. Morgan)

<u>1976</u>

Victor K. Barwell,
Numerical Solution of Differential-Difference Equations
(Supervisor: J.D. Lawson)

Lewis Denver Baxter,
The Complexity of Unification
(Supervisor: T. Pietrzykowski)

Bui Ngoc Duong,
A Formalism for Expressing and Studying Programs and Their Computations
(Supervisor: E.A. Ashcroft)

J. W. H. Liu,
On Reducing the Profile of Sparse Symmetric Matrices
(Supervisor: J.A. George)

Claudio Lucchesi,
A Minimax Equality for Directed Graphs
(Supervisor: D.H. Younger)

Ignacio Mijares,
Structural Design of Logical Data Bases
(Supervisor: E.G. Manning & R.W. Peebles)

K. Vidyasankar,
Some Covering Problems in Directed Graphs
(Supervisor: D.H. Younger)

<u>1975</u>

Ayed Amir Bukhari,
A Relational Data Base Model
(Supervisor: J.G. Linders)

Walter F. Finden,
Some Numerical Procedures for Solving Systems of O.D.E.'s Containing a Small Parameter
(Supervisor: J.D. Lawson)

Wilf Roger Lalonde,
Practical LR Analysis of Regular Right Part Grammars
(Supervisor: K. Culik)

Anthony Lavia,
Perturbation Methods in the Topological

Design of Computer Communications Networks
(Supervisor: E.G. Manning)

Jaroslav Opatrny,
Parallel Rewriting Systems
(Supervisor: K. Culik)

James A. Smith,
Automated Generation of Logic Diagrams
(Supervisor: J.G. Linders)

David A. B. Swayne,
Computation of Rational Functions with Matrix Argument with Application to Initial-Value Problems
(Supervisor: J.D. Lawson)

W. M. L. VanCleemput,
Mathematical Models and Algorithms for the Circuit Layout Problems
(Supervisor: J.G. Linders)

Arndt von Staa,
Data Transmission and Modularity Aspects on Programming Languages
(Supervisor: W.M. Gentleman)

<u>1974</u>

Julián A. Aráoz Durand,
Polyhedral Neopolarities
(Supervisor: J. Edmonds)

Hendrik Jacobus Boom,
Optimization Analysis of Programs in Languages with Pointer Variables
(Supervisor: D.D. Cowan)

William S. Bowie,
Towards a Distributed Architecture for OS/360
(Supervisor: J.G. Linders)

Ying-Tung Chen,
Iterative Methods for Linear Least Squares Problems
(Supervisor: W.M. Gentleman)

Sergio E. R. de Carvalho,
Design of Interrelated Lexical and Syntactical Analyzers
(Supervisor: K. Culik)

Jean-Louis Houle,
A Formal Language for the Description Modelling and Subsequent Realization of Digital Systems
(Supervisor: J.G. Linders)

David C. Lam,
Implementation of the Box Scheme and Model Analysis of Diffusion-Convection Equations
(Supervisor: R.B. Simpson)

Terence Chun-Yat Lau,
A Class of Approximations to the Exponential Function for the Numerical Solution of Stiff Differential Equations
(Supervisor: J.D. Lawson)

Luigi Logrippo,
Renamings in Parallel Program Schemas
(Supervisors: E.A. Ashcroft & J.A. Brzozowski)

Gabriele Ricci,
The Abaci (on the Representation of Data and Representation Dependent Algorithm)
(Supervisor: K. Culik)

James W. Welch,
Towards the Effective Implementation of Descriptive Storage
(Supervisor: J.W. Graham)

<u>1973</u>

R. J. Collens,
A Computer System for Balanced

Incomplete Block Designs
(Supervisor: D.D. Cowan)

R. B. Maguire,
Methods for Producing Visual Displays of Linear Graphs
(Supervisor: D.D. Cowan)

R. L. Probert,
On the Complexity of Matrix Multiplication
(Supervisor: P.C. Fischer)

L. D. Rogers,
Optimal Paging Strategies and Stability Considerations for Solving Large Linear Systems
(Supervisor: W.M. Gentleman)

L. V. Saxton,
Input-Ouput Conventions and the Complexity of Transductions
(Supervisor: P.C. Fischer)

1972

M. S. Doyle,
The Effects of Program Behaviour on the Design of Storage Hierarchies
(Supervisor: J.W. Graham)

P. S. Kritzinger,
An Approach to the Optimization of Direct-Access Merge Performance
(Supervisor: J.W. Graham)

I. Simon,
Hierarchies of Events with Dot-Depth One
(Supervisor: J.A. Brzozowski)

D. M. Symes,
The Extension of Machine-Independent Computational Complexity Theory to Oracle Machine Computation and to the Computation of Finite Functions
(Supervisor: P.C. Fischer)

W. S. Tan,
Numerical Solution of Second Order Integro-Differential Equations
(Supervisor: C.F. Fisher)

T. C. Wilson,
A Graph-Theoretic Approach to Some Problems in Compiler Code Optimization
(Supervisor: D.D. Cowan)

1971

D. K. Banerji,
Residue Arithmetic in Computer Design
(Supervisor: J.A. Brzozowski)

F. H. Chipman,
Numerical Solution of Initial Value Problems Using A-Stable Runge-Kutta Processes
(Supervisor: J.D. Lawson)

A. R. Conn,
A Gradient Type Method of Locating Constrained Minima
(Supervisor: T. Pietrzykowski)

G. R. Custeau,
Aspects of the Design of Maintainable Computers
(Supervisor: E.G. Manning)

I. P. McWhirter,
Substitution Expressions for Context-Free Languages
(Supervisor: J.A. Brzozowski)

D. E. Morgan,
Representation and Analysis of Computer Systems and Processes
(Supervisor: E.G. Manning)

J. C. Warkentin,
Small Classes of Recursive Functions and Relations

(Supervisor: P.C. Fischer)

C. R. Zarnke,
A Software Development Language
(Supervisor: D.D. Cowan)

<u>1970</u>

D. J. Cohen,
Computer Based Fault Analysis of Digital Systems
(Supervisor: E.G. Manning)

B. Nash,
Context-Free Parallel Levelled Languages
(Supervisor: J.A. Brzozowski)

<u>1969</u>

B. L. Ehle,
On Padé Approximations to the Exponential Function and A-Stable Methods for the Numerical Solution of Initial Value Problems
(Supervisor: J.D. Lawson)

Appendix E

Graduate Courses in Computer Science

(1991/92)

CS 642	Principles of Programming Languages
CS 644	Compiler Construction
CS 646	Software System Design and Implementation
CS 648	Introduction to Database Management
CS 650	Computer Architecture
CS 652	Real-Time Programming
CS 654	Distributed Systems
CS 657	Queueing Models: Analysis, Simulation and Computer Applications
CS 658	Design of Microprocessor-based Systems
CS 662	Formal Languages and Parsing
CS 664	Computational Complexity Theory
CS 666	Algorithm Design and Analysis
CS 667	Introduction to Parallel Algorithm Design
CS 668	Program Verification
CS 672	Numerical Linear Algebra
CS 676	Numerical Solutions of Differential and Integral Equations
CS 678	Numerical Solution to Partial Differential Equations
CS 679	Splines and Their Use in Computer Graphics
CS 684	Introduction to Computational Linguistics
CS 685	Intelligent Computer Interfaces
CS 686	Introduction to Artificial Intelligence
CS 687	Introduction to Symbolic Computation
CS 688	Introduction to Computer Graphics
CS 689	User Interface Tools
CS 690	Literature and Research Studies

CS 740	Topics in Software Design
CS 742	Data Types and Programming Languages
CS 744	Topics in Compiler Construction
CS 746	Software Portability
CS 748	Topics in Data Bases
CS 750	Topics in Computer Systems
CS 754	Topics in Operating Systems
CS 755	Operating Systems Laboratory
CS 756	Computer Networks and Distributed Processing
CS 757	Modelling and Analysis of Computer Systems and Comm. Networks
CS 758	Topics in Parallel Computation
CS 760	Topics in the Theory of Computing
CS 761	Finite Automata
CS 762	Topics in Formal Language Theory
CS 766	Computational Complexity
CS 768	Topics in Program Semantics
CS 769	Topics in Switching Theory
CS 770	Topics in Numerical Analysis
CS 772	Topics in the Solution of Linear Equations
CS 774	Topics in Numerical Approximation
CS 776	Topics in the Solution of Differential Equations
CS 778	Topics in Surface Modelling Techniques
CS 779	Topics in Spline Theory
CS 786	Topics in Artificial Intelligence
CS 787	Topics in Symbolic Computation
CS 788	Topics in Computer Graphics

Appendix F

Research Activities in Computer Science & Faculty Computing Facilities

1992

Centre for the New Oxford English Dictionary, Text Research
text-dominated database management, information retrieval, document management

Computer Graphics
3-D modelling and spline descriptions of curves and surfaces, human-computer interaction, production of high quality digital images

Computer Networks and Distributed Systems
digital switching techniques, communication protocols, network performance evaluation, LANs, videotext systems, parallel architectures

Computer Systems Group
software engineering, portability, tagged documents and text databases, programming language design and implementation

Data Structures
design and analysis of algorithms, computational geometry, text processing, design of and evaluation of database systems including multi-media systems

Logic Programming and Artificial Intelligence
planning, search, knowledge representation, automated reasoning, natural language understanding and user modelling

Office Systems
multi-media (interfaces, structures, object management, access methods), optical disk-based architecture, distributed office systems

Scientific Computing
sparse matrix solution techniques, linearization methods for large nonlinear algebraic equations, finite element methods for PDEs, timestepping algorithms for DEs

Symbolic Computation

the Maple symbolic computation system

Theory of Computing

design and analysis of algorithmscomputational complexity, formal language theory, automata theory, structural complexity, VLSI theory

VLSI

silicon compilation, hardware description languages, hardware modelling, asynchronous networks, fault detection

Institute for Computer Research

consists of thirteen federated research groups including Computer Communications, Graphics, Computer Systems Group, Data Encryption, Artificial Intelligence, Symbolic Computation and VLSI.

Computing Facilities:

At present, the systems operated by the Math Faculty Computing Facility (MFCF) include several VAXes (an 8650, 11/785s), two DEC 5500s, a MIPS M/6280 and eleven microVAX IIs), a network of seventeen SUN 3/50's workstations paired with seventeen NCD X-window terminals connected to a SUN 3/280 file server, one SUN 3/160, one MIPS M/2000, one MIPS M/120, one Sequent S81 and seven IRIS colour graphics workstations. All run some variant of Berkeley UNIX. Software includes several text editors, electronic mail, user-controlled archiving, plotting, text formatting, typesetting and a wide range of general and special purpose languages.
Communication between machines is primarily implemented with Ethernets. MFCF participates in the UNIX-based USENET providing news and computer conferencing.

Local users may access machines via a campus-wide Sytek local area network, Gandalf modems or dial-up phone lines, or datapac.
MFCF operates a number of labs including a real-time programming lab, a microsystems lab and a Macintosh II lab for first year computer courses. The Computer Science department also operates a Microsystems Lab in support of microprocessor-based designs.

Appendix G

Faculty Members in the Department of Computer Science[a]

1967/68

Professors:

Janos Aczel	PhD (Budapest), Habil.DSc (Hung. Ac. of Sc.)	
J. Wesley Graham	BA, MA (Toronto)	Director of Computing Centre
H. Haruki	PhD (Osaka)	
Mike A. McKiernan	MA, PhD (IIT)	

Associate Professors:

Donald D. Cowan	BASc, MSc, PhD (Waterloo)	AA/CS Department Chair
J. Douglas Lawson	BASc, PhD (Waterloo)	

Assistant Professors:

P.C. Jennings	Cross-appointed (Psychology)
John. C. Wilson	BASc, PhD (Waterloo)

Lecturers:

Peter C. Brillinger	BSc, MA (Waterloo)
Paul H. Cress	BSc, MSc (Waterloo)
Paul H. Dirksen	MSc (Waterloo)
Jan G. Kent	MSc (Oslo)
B.H. Ehle	BA, MS (Stanford)

Visiting Professor:

S. Golab	PhD, D.Habil (Krakow)

1991/92

Professors:

Richard H. Bartels BS, MS, PhD (Stanford) Cross-appointed (Appl. Math)

Research interests include curve and surface representation and manipulation, data fitting, and computer aided geometric design.

Janusz (John) A. Brzozowski BASc, MASc, MA, PhD (Princeton)

Research interests include asynchronous circuits, delay-insensitive design, testing, automata, formal languages.

Stavros Christodoulakis BSc, MSc, PhD (Toronto)

C.J. Colbourn BSc, MMath, PhD (Toronto) Cross-appointed (C&O)

---

[a] In the following listings, only the location of the *highest* degree is identified.

Donald D. Cowan BASc, MSc, PhD (Waterloo)
Research interests include software engineering, graph theory, computer networks, text processing and computer-based tools for education.

Keith O. Geddes BA, MSc, PhD (Toronto) Assoc. Dean /Grad Studies
Research interests include algebraic algorithms, systems for symbolic computation, scientific computation, numerical approximation, and the symbolic/numeric interface. Co-director: Symbolic Computation Group.

J. Alan George BSc, MSc, PhD (Stanford) VP Academic and Provost
Research interests include large-scale scientific computation, numerical algebra, and parallel computation.

J. Wesley Graham BA, MA (Toronto)
Research interests include software engineering and databases.

Per-Åke (Paul) G. Larson BEcon, MBA, PhD (Abo Sw. U) Department Chair
Research interests include database systems, file organization, data structures, and analysis of algorithms.

Ronald C. Mullin BA, MA, PhD (Waterloo) Cross-appointed (C&O)

J. Ian Munro BA, MSc, PhD (Toronto)
Research interests include data structures, analysis of algorithms, computational complexity, and text dominated data bases.

R. Bruce Simpson BSc, MASc, PhD (Maryland) Cross-appointed (Appl. Math)
Research interests include scientific computation, mathematical software, and the solution of partial differential equations.

P. Thagard BA, MA, MS, PhD (Toronto) Cross-appointed (Philosophy)

Frank W. Tompa ScB, ScM, PhD (Toronto) Director: New OED Centre
Research interests include data structures and databases, with applications to machine-readable dictionaries and hypertexts. Director: Centre for the New Oxford English Dictionary and Text Research.

Scott A. Vanstone BMath, MMath, PhD (Waterloo) Cross-appointed (C&O)

Johnny W. Wong BS, MS, PhD (UCLA) Associate Provost, Comp.&Inf.Systems
Research interests include computer networks, communication protocols, queueing theory, and performance evaluation.

Derick Wood BSc, PhD (Leeds)
Research interests include document processing, computational geometry, data structures and design and analysis of algorithms.

<u>Associate Professors:</u>

John C. Beatty AB, PhD (Berkeley)
Research interests include documentation graphics, the modeling of sculpted surfaces, raster graphics, color, and user-interfacing techniques.

James P. Black BSc, Dipl. d'Ing, PhD (Waterloo) MFCF Director, Assoc. Dean/Computing
Research interests include distributed debugging, network management, distributed systems, reliability and performance analysis. Director of the Shoshin distributed system group.

Forbes J. Burkowski BSc, MMath, PhD (Waterloo) Assoc. Chair/Undergrad

Research interests include text retrieval systems and computer architecture.

Edward P.F. Chan BSc, PhD (Toronto)

Research interests include deductive and object-oriented databases, knowledge-based management systems, relational database theory, schema analysis and design, logical database.

Robin Cohen BA, MSc, PhD (Toronto)

Research interests include artificial intelligence, natural language processing, computational linguistics, and social issues in computing.

Gordon V. Cormack BSc, MSc, PhD (Manitoba)

Research interests include data structures, programming languages, compilers, and concurrent programming.

William B. Cowan BSc, PhD (McGill) Cross-appointed (Psych, Fine Arts)

Research interests include user interface design, interactive real-time systems, colour and attention in graphical interfaces, and image quality and standards.

V. Arnie Dyck BMath, MMath (Waterloo)

Peter Forsyth BSc, MSc, PhD (Western) Assoc. Chair/Grad

Ming Li MS, PhD (Cornell)

Research interests include machine learning and computational complexity.

Farhad Mavaddat BSc, PhD, DIC (Imperial College)

Research interests include large scale digital design, synthesis verification, silicon compilers, VLSI architectures, design automation, and computer systems.

Jeffrey O. Shallit PhD (Berkeley)

Research interests include number-theoretic complexity (the complexity of algorithms for testing primality, integer factorization, discrete logarithm, etc.), formal languages, and automata theory, applicative programming languages (especially APL).

David J. Taylor BSc, MMath, PhD (Waterloo)

Research interests include debugging in distributed systems, monitoring and control of computer networks, atomicity and concurrency in distributed systems, fault-tolerant software, data structures for fault-tolerant systems.

Assistant Professors:

Fahiem Bacchus BSc, MSc, PhD (Alberta)

Research interests include the application of probability and statistics to AI, non-monotonic inference, logics for representation and reasoning, automated diagnosis, automated reasoning.

Peter A. Buhr BSc, MSc, PhD (Manitoba)

Research interests include programming languages, specifically polymorphism, concurrency, and persistence.

Johnathan F. Buss BSc, PD (MIT)

Research interests include computational complexity and parallel computation.

Michael H. Coffin BS, MS, PhD (Arizona)

Research interests include programming language design and implementation, parallel and distributed programming, operating systems, scientific and mathematical computation.

Chrysanne DiMarco BSc, MSc, PhD (Toronto)

Research interests include computational stylistics, pragmatics, computational linguistics, artificial intelligence, machine translation, second-language learning, language generation.

Dominic Duggan BComm, MSc, MS, PhD (Maryland)
Research interests include Type systems for software reuse. Specification and verification for programming-in-the-large, Meta-programming languages, semantics-based programming environments.

Jo C. Ebergen MSc, PhD (Eindhoven)
Research interests include programming methodology, parallel computations, and VLSI design.

George Labahn BSc, MSc, PhD (Alberta)
Research interests include algebraic algorithms, symbolic computation, rational approximation.

Anna Lubiw BSc, MMath, PhD (Toronto)
Research interests include computational geometry and combinatorial optimization.

Naomi Nishimura BS, MSc, PhD (Toronto)
Research interests include computational complexity, algorithm design and analysis, and data structures.

John L. Ophel BSc, PhD (Austr. Nat.)
Research interests include programming languages, compilers and programming environments.

Prabhakar Ragde BMath, PhD (UC Berkeley)
Research interests include computational complexity, theory of parallel computation, randomized algorithms, probabilistic analysis of algorithms, combinatorial optimization.

Hans-Peter Seidel BS, MS, PhD, Habil (Tubingen)
Research interests include computer aided geometric design, freeform curves and surfaces, geometric modeling, splines, computer graphics.

Wendy L. Seward BSc, MSc, PhD (Toronto) Cross-appointed (Appl. Math)
Research interests include numerical solution of time-dependent differential equations and development of mathematical software.

Wei Pai Tang BS, MS, PhD (Stanford) Cross-appointed (Appl. Math)
Research interests include large scale scientific computations, parallel algorithms, advanced parallel computer architecture, numerical linear algebra.

John H. Vellinga BA, MA (Waterloo)

Grant E. Weddell BSc, MSc, PhD (Toronto)
Research interests include compilation and performance issues for object-oriented data models, physical design and query optimization, and complex object encoding and indexing.

Qiang Yang BS, MS, PhD (Maryland)
Research interests include artificial intelligence, A.I. planning, heuristic search, knowledge representation, expert systems and automated design and manufacturing.

Research Assistant Professor:

Neil Coburn BMath, MMath, BEd, PhD (Waterloo)
Research interests include parallel database theory and systems, query optimization, constraints on complex objects, physical database design, and heterogeneous database theory and systems.

Lecturers:

B.W. Becker	BA, MMath (Waterloo)
B.D. Pawliszyn	BSc, BEd, MS (Utah)
Ann E. Pidduck	BASc, MASc (Waterloo)

Adjunct Faculty:

G. Barahona	Ing.Mat., DIng (Grenoble)
K.S. Booth	BS, MA, PhD (Berkeley)
A.R. Conn	BSc, MSc, PhD (Waterloo)
S. Christodoulakis	BSc, MSc, PhD (Toronto)
W.M. Gentleman	BSc, MA, PhD (Princeton)
R.N. Goldman	BS, MA, PhD (Johns Hopkins)
Gaston H. Gonnet	CprU, MMath, PhD (Waterloo)
J.H. Johnson	BMath, MMath, PhD (Waterloo)
J. Pachl	RNDr, PhD (Waterloo)
M.W. Sherk	BSc, MSc, PhD (Toronto)
J. Slonim	BSc, MSc, PhD (Kansas State)
T. Stepien	BMath, MMath (Waterloo)
D.W. Swayne	BSc, MA, PhD (Waterloo)
P. Van Arragon	BS, MMath, PhD (Waterloo)
M. Wein	BEng, MSc, PhD (McGill)

Appendix H

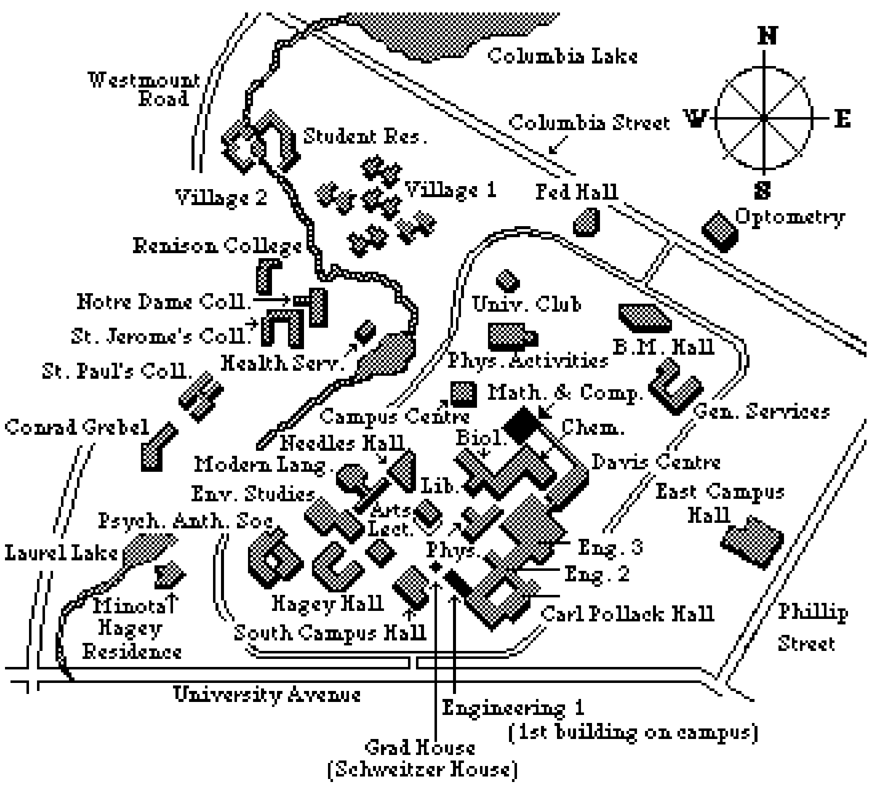
Chairmen of Computer Science

(1967 - 1992)

1967 - 1972	Donald D. Cowan	BASc (Toronto), PhD (Waterloo)
1972 - 1975	Patrick C. Fischer	MBA. (Michigan), PhD (M.I.T.)
1975 - 1978	J. Douglas Lawson	BASc. (Toronto), PhD (Waterloo)
1978 - 1983	Janusz A. (John) Brzozowski	MASc. (Toronto), PhD (Princeton)
1983 - 1987	R. Bruce Simpson	MASc. (Toronto), PhD (Maryland)
1987 - 1989	Janusz A. (John) Brzozowski	MASc. (Toronto), PhD (Princeton)
1989 - 1992	Per-Åke G. (Paul) Larson	BEcon, MBA, PhD (Abo Swedish U.)

Waterloo Campus Map

1992



Appendix I

Some Conferences held at Waterloo

October 15-17, 1969
Tenth Annual Symposium on Switching and Automata Theory
local arrangements: J.A. Brzozowski

Aug. 15-17, 1977
Conference on Theoretical Computer Science
program coodinators: E.A. Ashcroft, J.A. Brzozowski, J.I. Munro

May 10 & 11, 1979
Computer Science & Statistics: 12th Annual Symposium of the Interface
local arrangements: Jane F. Gentleman

April 2-4, 1984,
Third ACM SIGACT-SIGMOD Symposium on Principles of Database Systems
local arrangements: P-Å. Larson

July 9-14, 1984.
Householder-Gatlinburg Symposium on Numerical Algebra IX
(International Symposium held every three years, alternately North America and Europe.)
local organizer: J.A. George

July, 1986
SYMSAC '86: ACM Symposium on Symbolic and Algebraic Computation
local arrangements: K.O. Geddes

June, 1987
The Third ACM Symposium on Computational Geometry
local arrangements: Derick Wood

March 29,30, 1990
SIAM Regional Meeting
Sponsors: SIAM Great Lakes Chapter ITRC
local organizer: Richard Bartels

May 30 - June 1, 1990.
Databases Directions for the 90's
general and program chairman: D. D. Cowan

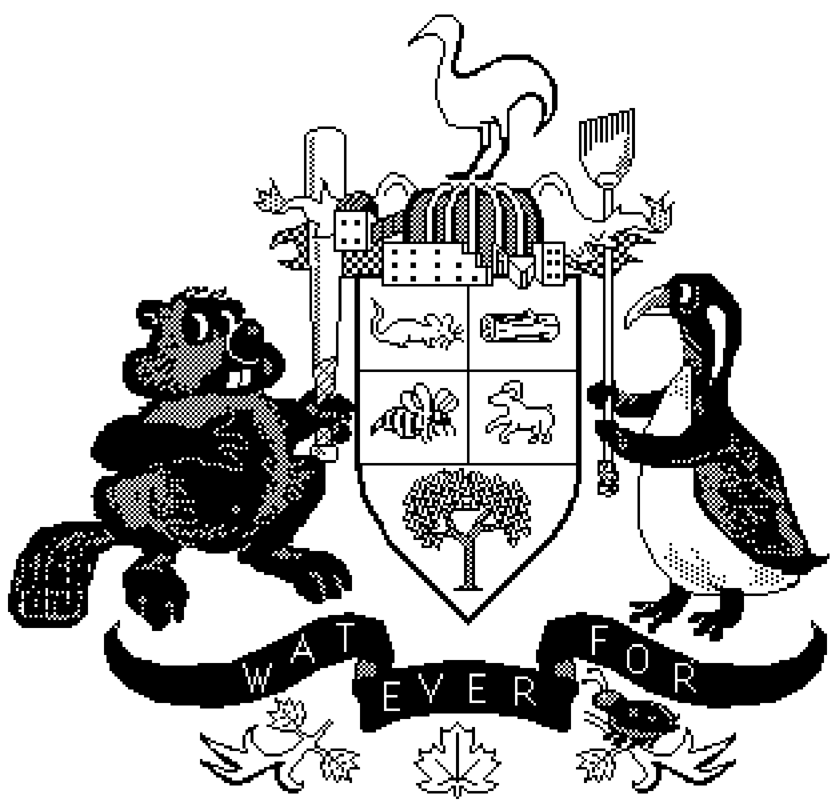
Appendix J

Coat of Arms

The student-designed coat of arms features (among other things):

The Departmental Mascot
The Logic Programming and Artificial Intelligence Mascot
The Computer Graphics Mascot
The Symbolic Computation Group Logo
some popular departmental recreation (?)
a graphics mouse
B-splines & ***B***-trees
Random Access Memory
a binary tree
a *log* ... as in *n log n* algorithm performance
a programming bug
the William G. Davis Centre
WAT FOR EVER (... or is it WAT EVER FOR ?)

and what else?



Appendix K

"Memories"

by Ian! D. Allen[a]

I was going to come to UW as a student in Integrated Studies in 1972. I was interviewed and accepted, but decided to stay out of school for a couple of years and work. When I finally decided to go to University in 1974, I switched disciplines and applied to take Psychology. I thought: *I can probably take the same courses in Arts that I could take in I.S., and at least in Arts I get a degree that doesn't require an explanation wherever I go!*

So, I started at UW in September 1974 as an Arts student in Psychology. I took my first-ever Computer Science course as an elective that fall from John Vellinga: Math 122a. During the two days after I got my first computer account number on the DEBUG system (it was a Wednesday), I had a fantastic amount of fun keypunching WATFIV programs and running jobs. Structured WATFIV came out that term, and I got annoyed comments from the confused Math 122a tutors when I started using it in my assignments. I decided to take as many elective CS courses as I could.

The next term we started learning machine code using WATIAC/Spectre. Spectre was a pseudo-assembler for a hypothetical language running under a simulated virtual machine, and the vagueness of it all was a continual source of amusement to me.

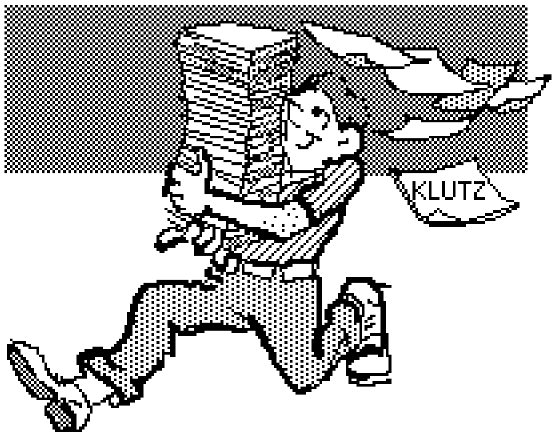
I learned that by punching a "G" in place of the "D" of the account number, I could get extra CPU time. We all did this.

Somehow, I found the WITS (Waterloo Interactive Terminal System) terminal room, and discovered my DEBUG account number worked there too. No more punch cards! I could write and submit jobs from the comfort of a CRT. To this day, due to its flexibility with column and line ranges, the WITS editor is the only one I know of that could display a file both upside-down *and* backwards at the same time. WITS also neatly solved the problem of people typing when the computer wasn't ready by actually physically locking the keyboard when it didn't want to listen to you.

---

[a] Research Project Manager, Computer Graphics Lab. This appendix is the content of e-mail messages I received from Ian while he was in Ottawa and I was writing this CSH. (PJP)

c=1000



My lack of knowledge of IBM JCL once led me to think that c=1000 on the //JOB card would give me more core memory. When I went down later to the I/O room to pick up my output, it wasn't in the usual folder under my account number. I discovered a foot-high stack of paper in the output boxes, tastefully annotated on the top with the word **KLUTZ** in large, bold black marker. That stack of paper was 1,000 copies of my JCL ... 1,000 copies of my job that, of course, had failed due to insufficient core.

I discovered that as an Arts student taking CS elective courses, I was not subject to any of the rules and regulations of the CS faculty. I was actually allowed to take *both* business and non-business CS courses for credit; something my CS friends could not do.

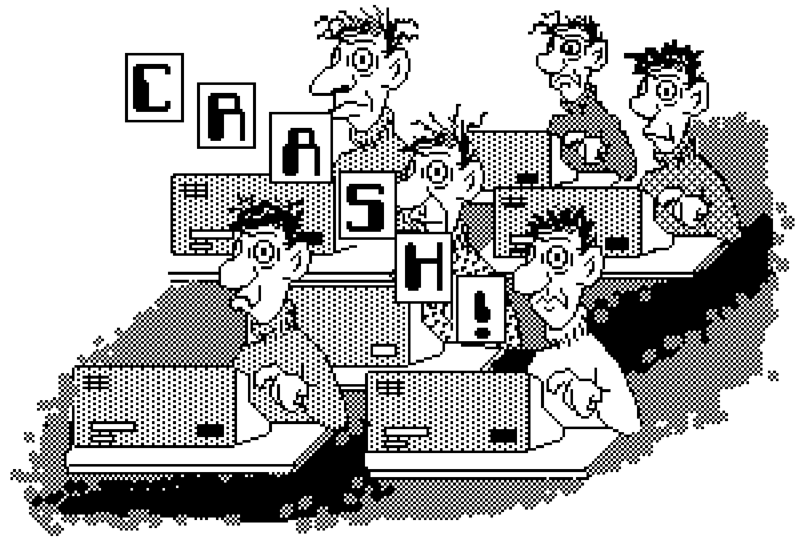
Somewhere in my memorabilia I have a pink punch card signed by J.W. Graham that entitled me to be one of the first Guinea Pigs for an introduction to WIDJET for my Math 132 course on File Processing and WATBOL. Lucky for me, I discovered the Honeywell the next term and didn't have to spend much time on WIDJET. *"YOU ARE QUEUE POSITION 158..."*

The Honeywell 6050 timesharing system is where I really grew up.

Ciaran O'Donnell was one of the more infamous students of the day, and I remember signing on to the Honeywell one evening to be greeted by a login banner that said "Ciaran's 6050". The system prompt was changed to say "WHAT?" after every command. New mail produced "** BEER WAITING", and entering the mail system produced the informative "12 FROGS IN BEER CAN".

Life is Cruel

To the Honeywell was connected a room full of IBM 2741 teletypewriters. The sound in this room was deafening. Dozens of "dancing-ball" type-heads were bashing out words at the amazing speed of about 13 characters per second.



Sometimes, the entire room would suddenly stop dead; an ominous silence would replace the din, punctuated only by the coordinated gasp of two dozen people all holding their breath simultaneously. Then, all two dozen teletypewriters would spew out down the page in perfect rhythm that five-letter word we all came to know and fear:

C
R
A
S
H

indicating that the Honeywell was dead, and our work was forever lost.

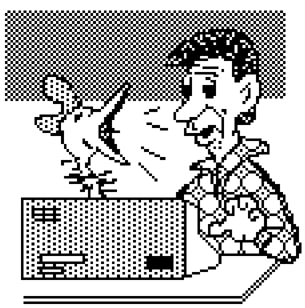
This is much more picturesque than almost anything I've seen since, except perhaps the famous last words of most Unix systems: Life is Cruel. It certainly has "SYSTEM ABEND 80A" beat, and nothing can compare with the drama of that deadly silence, that agonizing pause, then the terrible thunder of two dozen machines typing CRASH! in perfect harmony.

M.H. van Emden taught the first CS course wherein I hadn't a clue what was going on for the first month. My partner and I eventually caught on, and we split the work load: with my Arts background in writing essays, I wrote the specifications and the manual; he wrote the software. We "wow'ed" the TA's and instructor with brilliant documentation, artfully text-formatted using the Honeywell's ROFF formatter. It looked gorgeous, and said almost nothing specific. I think we got away with it. With one foot in Arts and another in Computer Science, I found I could impress each faculty with skills learned in the other. CS professors were always pleased with the fact that I could write reasonable English. The Arts faculty were impressed with essays that had been run off with justified margins, running titles, and other tricks of the ROFF text formatter.

When the Photon Econosetter phototypesetter arrived, and my essays started looking like printed articles, this was sufficiently new that my papers would get circulated around the Psychology Department more for looks than content.

Taking the cream of CS courses while enrolled in Arts, I found I had hours of extra time that I could spend at the terminals. I had an unreasonable amount of fun programming, almost to the exclusion of basic survival needs such as eating and sleeping. (I once arrived back at my residence floor in Village One after a long afternoon at the terminals and was greeted with "What are you doing here? Didn't you say you had to be in Toronto for your girlfriend's dinner party?". I *did* say that; in the middle of programming, I had simply forgotten to leave.)

Flying Turkey



The IBM 2741 teletypewriters had at least one tiny flaw; they didn't quite know how long it took to bring the type ball back to the left margin after being sent a bunch of TAB characters. People would prepare e-mail messages with a dozen TABs followed by a carriage return and the word "YEKRUT". The TABS invariably sent your typeball all the way off the right margin of the page, and the "YEKRUT" was typed as the ball flew back to the left margin. Of course, "YEKRUT", when typed right-to-left,spelled "T-U-R-K-E-Y" on your paper. This sequence made a distinctive noise, so that everyone in the room knew you'd just been "TURKEY'd".

Fred Flintstone

Peter Fraser wrote his own version of the Bell Labs' QED tex editor. The prototype version was called "DUD", for what were at times reasonable reasons; however, the final name for the editor was "FRED". Steve Hayman (of Warriors' Band fame) immediately wrote a little editor script that he named "FLINTSTONE", and you executed it on the Honeywell by simply typing "FRED FLINTSTONE". It printed various alarming (but harmless) messages and pretended to disconnect your terminal.

Darth CRT

In the early days of the Computer Graphics Lab, Sylvia Lea took a picture of a mannekin wearing a Darth Vader helmet in front of our heavy black Evans&Sutherland MPS screen. Under the picture was the caption: "We specialize in Friendly User Interfaces for Unfriendly Users".

The TrackBall Cake

Dave Martindale (of the Computer Graphics Lab) was finally finishing his Master's Degree. His final project was a trackball controller. To celebrate this event, and as an excuse for a party, Jeanine Malito and some friends made him a 2/3 scale model of his project as a cake.

World's Most Expensive Alarm Clock

When computers got the ability to dial telephones, we threatened to abuse this capability by having them wake people up at night. As it turns out, the one time we actually programmed this to happen, to call and wake up our sysadmin early one morning for a plane flight, a typing mistake prevented the software from dialling the phone. Lucky for him, his alarm clock did the job. Moral: Don't use $800,000 worth of hardware where $12 worth will do the job.

Good Verbal Communication Skills

At one point my apartment on Erb St. had three Computer Graphics people in it. This required, of course, three telephone lines. Two of us were discovered at home one evening using the inter-terminal TALK program to communicate, when we were, in fact, in adjacent rooms in the same apartment. The proliferation of dial-out lines reached its peak with four phones in a house with three CS people. (One phone was the "house" line, reserved for incoming calls.) "The phone is ringing" became "a phone is ringing", followed by a scramble to find out whose it was. The principal disadvantage to this number of phone lines was the much higher Saturday morning hit-rate by people offering carpet cleaning.

Night PLATO

Being in Arts and also interested in computers, I spent a little time around the WATCHUM (WATerloo Computing in the HUManities) office in the basement of the Psychology Building. In one of their tiny offices sat a single 512 by 512 pixel orange gas plasma screen hooked up during off-hours to the PLATO system at the University of Illinois at Urbana. (We shared the line with the University of Toronto; they used it during the day.) PLATO was a Computer-Assisted Instruction machine with hundreds of interactive lessons on all kinds of topics, plus more interesting things such as DogFight, AirFight, and a multi-user chatting program named TALKO. The capabilities of this system put to shame anything I knew about at Waterloo. I managed to convince someone that I should be granted "AUTHOR" status on PLATO. This would give me access to the programming language features and allow me to create my own interactive "lessons".

I would sign out the PLATO key from the WATCHUM office and come back late at night when UofT turned the line over to us. I'd be there all night, typing away with hundreds of other users, most of whom were in Illinois. PLATO allowed terminal-to-terminal messages; something I didn't see again until a decade later when the Internet connected us all.

Unlike most games, the ones on PLATO were reasonably realistic. If you happened to pull back too far on the stick of your aircraft, the screen would clear, and a friendly "CONGRATULATIONS!" message would appear, explaining how you had exceeded the g-forces allowed for your aircraft and how it was now in pieces all around you. The 3-D StarTrek game had real warp speed, and if you sent your craft too far from the "Known Universe", you'd end up with ****'s in all your co-ordinates and couldn't get home.

by Jerry Bolce[b]

The Beginnings of JANET

---

[b] Department of Computing Services

It was the Christmas holiday of 1981 that I had an IBM PC to take home. At that time the Physics Department was looking to install a lab of personal computers for the students. With the arrival of the IBM PC, it became obvious that the PC was the best choice.

Since hard disks and printers would be too expensive to install in each machine, a network with a central server was required. But there were no networks, no hard disks, not even a network circuit board. Adrian Weerheim had worked with the IEEE-488 or Hewlett-Packard Instrument Bus as implemented on the Commodore Pet Computer. The HPIB was available as a three-chip set, so he proceeded to design and build our network circuit boards for the PCs.

In the meantime, Chuck Pilkington, a student, and I worked at designing and coding the software required for the server and workstations. Our scheduled implementation date was mid September 1982.

The tools were somewhat lacking. The IBM Technical Reference Manual had not been written yet; the best we could manage was to get a listing of the ROM. Hard disks were not yet available so we had to guess how they would be implemented. For languages, we had a cross-assembler that was still being developed by the Computer Systems Group. It seemed to have its own ideas of what instructions to generate. And we also had Microsoft Basic on the PC.

Over the summer Adrian produced a run of network boards while Chuck and I worked on the software. A hard disk still had not been delivered and design decisions had to made to code the network file system for the server. The hard disk finally arrived on the Friday of the Labour Day weekend. On Thursday students were using the lab.

And now came the hard part - what to call it. Everyone just referred to "Jerry and Adrian's network". Chuck needed a name for his work report and decided on ICON, standing for IEEE Computer Network. I didn't like the name, as the IEEE was only an expedient solution. The suggestions then went: If it is Jerry and Adrian's Network, just call it JANET. What about Chuck and the others to come? Then how about "Just Another Network?" OK, why not!

by Ross Brown [c]

... about the Photon, or, as we mathNEWS staffers came to know it because of its erratic behaviour, the Pho^{o}to$_{N}$.

Donated to the Math Faculty by the Toronto Star as a tax dodge (as legend has it), this noisome, shuddering beast was so inscrutable and capricious that it acquired the status of a fearsome god, the sort to which virgins were sacrificed, before there were phototypesetters. The mysterious hardware was operated by equally mysterious software, known only to a club of hand-taught slobs. Getting a galley sheet of clear, uncreased, unpied, horizontally oriented text was the sort of accomplishment you could take real pride in. When I riffle through my pile of back issues, it's the craftwork I remember most fondly, not the material. Shame on you, me, and the rest of us computer scientists for homogenizing this business so that anyone with a Mac and a laser can do it!

by Shirley Fenton[d]

Musings on CS 100

In 1985, Jay Black, Arnie Dyck and I embarked on an incredible journey in redesigning the introductory Computer Science course for non-Math students at the University of Waterloo, in March of that year. We realized that teaching these students to learn a basic understanding of using the computer through the traditional method of programming did not meet their needs. Our task was to have the new course developed and operational for 600 students by the Fall term. We had no hardware, no software, and no course material. In effect, we were “starting from scratch”!

Over the next six months, through team-work and a spirit of co-operation, this course was developed. Everyone pitched in: the course instructors created the course materials; Apple Canada and Microsoft provided the hardware and the software; the Computer Systems Group, especially Carl Durance, developed the MacJANET networking software; the Department of Computing Services installed the hardware.

This course was a dramatic shift from the past. We moved from a course based on programming to one centered on productivity tools. The computing environment we wanted was totally new. Based on leading-edge technology,

---

[c] BMath (CS) '84 , past editor, mathNEWS

[d] formerly with the Department of Computing Services and an Adjunct Assistant Professor in CS; now in Operations Analysis.

we moved from a character, terminal-based system to a window, icon-based system of networked microcomputers. As instructors with years of experience in Computer Science, we found we were teaching tools we had never used before.

In many ways, we operated under a "ignorance is bliss" syndrome. Any number of factors could have caused us to fail in this undertaking. The operation of the computer laboratory was critical for the success of this course. To start this course with a small group of students with all the uncertainties would have been challenging; to undertake the full-running of the course with the laboratory in operation at full capacity with 600 students was demanding. The tutors, such as Jack Rehder and Allison Brown to name only two, managed the computer labs with endless enthusiasm even under the initial adverse conditions. We had to solve many logistical problems ranging from how to schedule 3,000 student-hours of computer time per week, how to share several printers among 45 machines, to how and when to clean the mice.

This course has stood the test of time even though the world of computing has been changing rapidly. The course structure, content and operation is as appropriate today as it was seven years ago. The success of this course led the Computer Science Department in several new directions. It led in part to the redesign of the introductory course for Mathematics students. It led to the development of CS 100 for correspondence students. In future, we believe that our foundation work on CS 100 will lead to the enhancement of computer literacy among all our students as they prepare for their life at home or at work.

By this time (1992) about 10,000 students have taken CS 100!

by Erna Unrau[e]

Typing and Secretarial work

In the mid-60's, as staff, we had little contact with or use for the computer. Electric typewriters were relatively new and we were pleased to have them to do our work. By today's standards they were quite old-fashioned.

There was a fair amount of mathematical typing that was done by the Computing Centre staff. Our typewriters had what we called a 'key' - a 4

---

[e] Erna was on the Computing Centre staff, from 1965 - then the first AA/CS department secretary until 1970.

inch hinged element that struck the paper once for each letter we typed. There were two letters on each key (lower case and capital). As we needed to put mathematical symbols into our document, we needed to click out the original key and click in the math key, type the symbol, remove the math element and reattach the original, type a few words, and so on. In the late 1960's IBM developed the selectric typewriter with the new round typing elements - (which we called golf balls). They too had to be interchanged for mathematical typing and many a one flipped out of our hands and chipped when it landed and rolled on the floor.

Theses were typed by the secretarys in the 1960's. Typewriters did not have built-in memories. Since theses had to be letter-perfect, if a major error was made early on in a thesis and it was enough to change succeeding pages, all the pages would have to be retyped. Sometimes more than once.

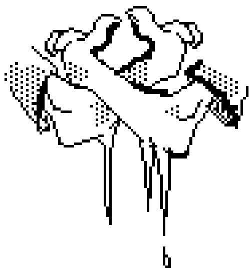
The waste of paper in those days was astounding compared to today's environmental awareness. When I started in the Computing Centre, (in 1965) the department's policy was that no letters or memos were to leave the office with a correction or erasure on it and each document was to be typed letter-perfect. That meant every time a typing error was made - out would go the original along with one, two or three carbons - into the waste paper basket. You started again with your original copy, a carbon, a piece of onionskin, a carbon etc. Another error? - out that would go and you started again. At these times there would be colourful language used. We got to be quite accurate typists. Even so, on those days when our fingers seemed to be 'all thumbs' our waste paper baskets were full to overflowing.

Staff made coffee in the lounge for the faculty. We even supplied cookies and each morning a staff member would go into the lounge and nicely put the cookies out on display, get the coffee going etc., it needed to be checked during the day and everything cleaned up in the evening. I think that practice was discontinued when the donations coming in no longer met the costs. Or perhaps someone decided that our time could be better spent doing more productive work.

Salaries

The late sixties were years where inflation in Canada was quite high and the salaries of faculty and staff tried to keep up with the cost of living. Some years we received raises of 13% (the inflation rate) plus the regular increments of 4 or 5%. Salaries could easily double in a few years.

Graduate Students

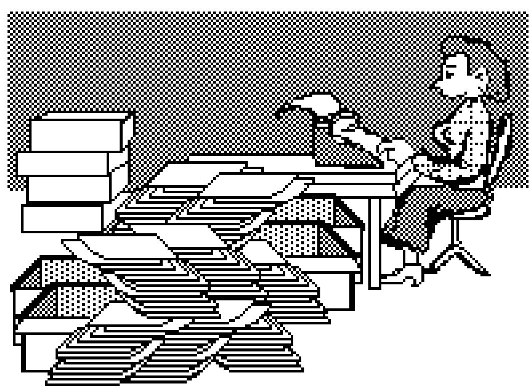


The computer science graduates were an interesting group of people coming from many countries. I especially remember one student who decided that he couldn't afford any shoes - so he walked barefoot everywhere. It took until February in the midst of a really cold and snowy spell, when he had frostbite on his feet, that he decided that perhaps he would need to get some shoes to walk outside.

Another student went back to India to get married after having studied in Canada for a few years. Upon his return he complained of the terrible heat rash from no longer being accustomed to the high temperatures.

Some of the graduate students really got into bridge playing in the 60's - playing during the days and many nights. One fellow got so involved that he neglected his graduate studies. His young wife decided to teach him priorities and refused to cook meals or wash any dishes until he quit playing cards and started studying. It took a few weeks. Once he realized that she could outlast him in stubbornness, he put on an apron and started washing dishes. He turned out to be one of the brightest graduate students at that time.

Computer Cards



There were rooms with boxes and boxes of computer cards, piled from floor to ceiling. - they all had to be kept in the correct order. Many overtime hours were spent on the key punch machine, punching holes into computer cards trying to set up an address list for Computer Science Days. A persons' address would have the name on one card and the address continued on the 2nd and 3rd card etc. - we really had to be careful not to get the cards out of order. There probably are a few stories still told of students standing in a long queue waiting to have their cards read, and by accident dropping the cards.

And what did we do with used computer cards? It was amazing at some of the decorations that people came up with. At Christmas time, wreaths were hung on many a door made up of stapled-together, spray-painted cards.

Computers

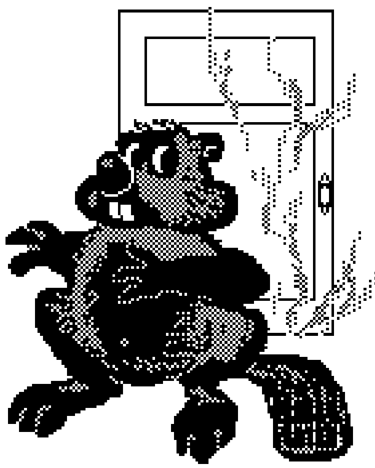
Staff were allowed to admire the computers from afar, and we, as well as visitors, were quite impressed with the sheer size (the bigger the more impressive) and speed, and enjoyed any presentation that the computer staff made.

In 1968 when we had moved into the new building there were many visitors admiring the huge red computer room on the second floor. Some tour groups were allowed into the room. The programmers enjoyed entertaining the visitors with the 'musical' printer. Depending on the number of characters per line, there was a variation in sound as the printer printed a document - fewer characters had a higher sound, more characters, lower etc. Someone had written a computer program that when printed out sounded like a song - I believe it was 'O Canada'. The visitors were really impressed. I think it cost $50 to run the song one time so it wasn't done too often and only when the supervisors weren't looking.

It was 1970 before the staff started using a computer on a limited basis to input papers. We started using it with trepidation, thinking that if we hit the wrong button we might blow something up. Our first attempts were quite involved - getting into the system, typing and printing. The system often crashed and we would have to call the programmers in the Computing Centre to get us back on-line, a few hours later we could get back into the document and start again. If we wanted to see our document we had to print the whole thing each time then run down to the 2nd floor to pick up the printout, make corrections, print the whole document etc.. We didn't use it very often - it was faster to use our typewriters.

Visitors/Faculty

These were years of growth not only in students but also faculty and staff. New courses were being established, research papers written, Computer Science Days, graduate work, NSERC applications. Waterloo's reputation in mathematics and computer science was growing and faculty and visitors arrived from many countries of the world - for short term or long-term assignments. It was like the United Nations - many languages spoken and everyone working together. They all had different stories to tell and the visitors enjoyed talking about their home countries. I remember Z. Dvoracek who had come from Czechoslovakia and how he enjoyed speaking about the mountains and countryside. He brought in a picture book of his beautiful homeland. During the Viet Nam war and the heightening Peace Movement, faculty and students (draft dodgers) arrived from the United States. Others came to Waterloo from Czechoslovakia in 1968 after the Soviet clampdown there. The U.S.S.R., Jugoslavia, Hungary, Poland, Australia, Nigeria, India, Italy, Brazil are some other countries which were represented.



Each time a visitor arrived it was time for a dinner or a wine and cheese party. I remember after one wine and cheese party there was a fair amount of cheese left. The person in charge had locked the cheese in his warm room over the weekend. We all knew where it was on Monday morning.

Meeting deadlines and crises seemed to be part of the job. So many situations were new and, along with computers breaking down, there always seemed to be too much work to do with the number of people there were to do it, and never enough hours in the day to get everything done. More and more staff was hired to keep up with the increase in faculty. The staff quickly increased from one secretary in 1966 to six in 1970. The staff too was international with secretarys often being the wives of graduate students.

One June we were trying to meet a deadline for an application for funding (I think it was to NSERC). Wes Graham was on a trip in Italy but needed to see the completed application before it was sent to Ottawa. We spent days in preparation, working evenings, typing and retyping. Finally at the last possible moment it was completed, copies were made and someone drove it to Toronto to be put on a plane to Italy (these were the days before Courier service). Luckily we had sent it to Ottawa without Wes's approval - I don't think he ever received it.

The staff learned how to type names from many countries. The Polish ones - typing all those consonants one after another - were a real challenge. We even had an 'absentminded professor' or two. Many of the professors had a wonderful sense of humour but no one matched K.D. Fryer for being able to say the right words at the right time to get everyone laughing. I remember one Professor exclaiming one day, with envy, what a special gift Professor Fryer had for teaching.

Mutt and Jeff?

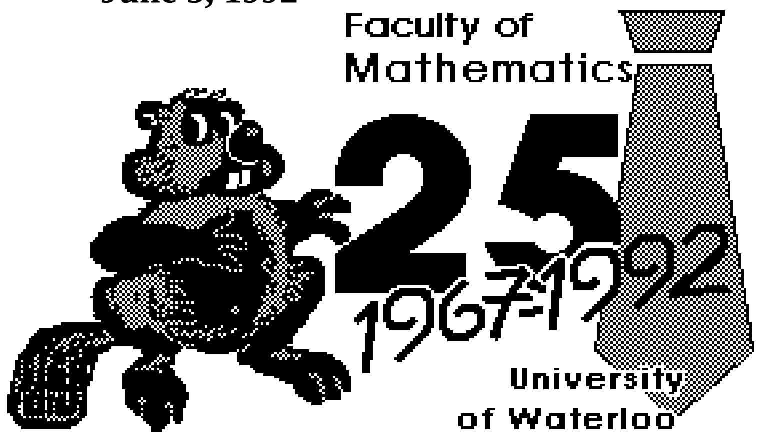
J.W. Graham and D.D. Cowan were together a lot those first years as the Computing Centre emerged into two departments. We always knew where Wes was when we heard his booming laugh echoing down the Physics Building hallway. When Don and Wes walked down the hall together, Wes who is quite tall and thin, seemed to be developing a permanent shoulder stoop as he hunched forward so he could talk with Don (who was not so tall or thin). Don had very straight posture.

I remember on Don's first trip to Brazil, how much he appreciated letters saying what we were doing in Waterloo. John Vellinga was looking after the Cowans' house during his leave and we reported that the fish were lonely, missed the Cowans and really looked happy when John arrived to feed them.

After the straight hallways of the Physics Building the maze of hallways on the 5th floor of the new Math and Computer building was very confusing. The staff and faculty soon learned the system but it seemed some students spent days walking around and around looking for a certain professor's room or trying to find their way out. This reminded us so much of a popular song of the day of someone lost in a subway system going in circles forever, with the chorus being *"Well, will he ever return? No he never returned ..."*

The FASS evenings also need to be mentioned as part of university life. Since the university was so much smaller most faculty and staff knew the whole campus and FASS evenings were a popular time where we could all laugh together at the foibles of the faculty and administration. Tickets were really hard to come by in those days. Students would line up at 4 a.m. in order to be first for tickets which were often sold out by 6 a.m.

Appendix L
25th Anniversary Research Day
June 5, 1992



In connection with the celebration of its 25th anniversary, the Department of Computer Science, University of Waterloo, cordially invites you to attend a special **Research Day** on **Friday June 5, 1992**. Seven of the Department's distinguished graduate alumni will present a series of research seminars. Admission is free.

For additional information please contact:
Linda Norton,
phone : (519)-888-4464
e-mail : lcnorton@water.waterloo.edu

Program

Scalable Imaging Architectures
Dr. Richard J. Beach,
BMath, MMath, PhD (Waterloo '71, '74, '85)
Principal Scientist and Lab Manager,
Electronic Documents Laboratory,
Xerox PARC, Palo Alto, California, USA

Inheritance in Object-Oriented Languages
Dr. Thomas A. Cargill,
BSc (Reading '73), MMath, PhD (Waterloo '75, '80)
Consultant, Boulder, Colorado, USA

Dissemination-Oriented Computer Communication
Prof. David R. Cheriton,
BSc (U.B.C. '73), MMath, PhD (Waterloo '74, '78)
Computer Science Department, Stanford University,
Stanford, California, USA

Large-Scale Nonlinear Constrained Problems
Dr. Andrew R. Conn,
BSc (Imperial College '67, MSc (Manitoba '70), PhD (Waterloo '71)
Staff Research Member, Thomas J. Watson Research Center,
Yorktown Heights, New York, USA

Lower Bounds for Fun and Profit
Prof. Faith E. Fich,
BMath, MMath (Waterloo '77, '79), PhD (UC Berkeley '82)
Department of Computer Science,
University of Toronto, Toronto, Ontario, Canada
presently : Visiting Scientist,
Laboratory for Computer Science,
Massachusetts Institute of Technology,
Cambridge, Massachusetts, USA

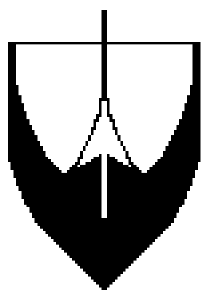
Research problems in computational biochemistry
Prof. Gaston H. Gonnet,
Cpr U (Uruguay '73), MMath, PhD (Waterloo '75, '77)
Informatik, Institute for Scientific Computation,
E.T.H. Zentrum, Zurich, Switzerland

Formal Hardware Verification:
Past, Present, and Future
Prof. Carl-Johan H. Seger,
Fellow, Advanced Systems Institute,
MASc (Chalmers '85), MMath, PhD (Waterloo '86, '88)
Department of Computer Science,
University of British Columbia,
Vancouver, British Columbia, Canada

Appendix Z

the University Coat of Arms

Although this has little to do with a *History of Computer Science at Waterloo*, I can't resist telling the story.



In the early 1960's, while a graduate student at the University of Illinois, I received a curious letter from Ralph Stanton. Accompanying the letter was a diagram of a crest, incorporating a stylized **W**. Without further explanation, Ralph asked me to *"draw about the **W** a humorous Ubangi woman with pendant breasts congruent to the **W**, and to have in her hand a spear with its point echoing the **W** motif"* ... or words to that effect.

I did as he asked, and later (perhaps ten years later!) discovered that the above crest had been considered for the University of Waterloo coat of arms. Ralph had submitted to Senate a critical assessment of the proposal, in which he states (among other things!):

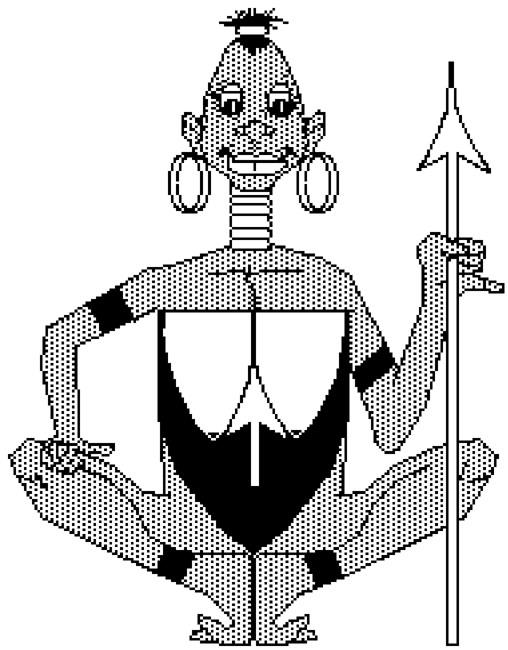
> The proposed coat of arms displays a certain sterile barrenness and lack of substance ... one should be able to display and wear (the coat of arms) with pride. I would personally feel apologetic and ashamed ...we would be the laughing stock among Canadian universities if we were to adopt a "University trademark" rather than a shield ... The proposal being presented, while not traditional for a <u>university</u>, is very similar and even commonplace as a company trademark. Similar pieces of design can be found upon all sorts of products ... I am sure it would be acceptable to Wilson's Ginger Ale ...

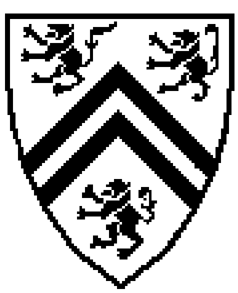
Ralph's submission to Senate then displays a number of industrial and commercial logos including ALCAN, JOHNSON's WAX, ALCOA, MOTOROLA and, of course, embedded among these, the proposed UW crest. The final appeal reads:

> The preceding submission and displays are made with the hope that the Senate will see fit to reject the proposed coat of arms.

The final page of the submission has, once again, the proposed coat of arms, with the comment:

```
     A slightly modified version of the proposed university
coat of arms is displayed to illustrate the fact that any
symbol can be dressed up in laurel leaves.
```





The current crest (which reflects Ralph's traditional thinking) was eventually approved (although the rampant lions have since changed from the original, stiff, identical felines - to conform to those on the University mace).

Peter James Ponzo
May, 1992

Index