Andrew D. Todd

The Free Amazons of Cybernatia: Women Computer Programmers, 1960-1980

Andrew D. Todd
1249 Pineview Dr., Apt 4
Morgantown, WV 26505

adtodd@mail.wvnet.edu

Abstract:

Computer programming helped many women overcome the economic and social constraints they faced in twentieth century America, by admitting them to a technical profession. The first women computer programmers emerged with the first computers. The programming profession expanded rapidly and continuously. Women had less difficulty in becoming programmers than they did in entering other technical professions, and many were able to remain in the field in spite of motherhood. The programming mother was often able to design a system of programming according to her needs. Programming could expand numerically only by expanding conceptually, by broadening the range of things which were programmed. Women tended to play a considerable role in this redefinition-- for example, by advocating PC's in the 1970's. The public image of woman programmers, as portrayed in the trade press, gradually caught up with the facts by showing women as technical experts rather than decorative sex objects. It is argued that female programmers made these gains without employing the political strategies of rhetoric of mainstream feminism.

In the year 1960, the programming profession was expanding rapidly and continuously. The middle period of computer programming, from 1960 to 1980, was, above all else, the period of initial engagement with the external world. It was the period in which the number of computer programmers rose from a few tens of thousands to hundreds of thousands,[1] and the number of computers from the hundreds to the low millions. Programming went from an obscure, if esoteric, field of technology to an ascendant field, not yet predominant, but with the promise or threat of ultimate predominance visible.

Programming's ascendancy was based on a series of inventions which made computers far cheaper and more usable than they had

---

1. At the time of ENIAC, in 1943-44, there were, depending on definitions, perhaps as few as a dozen programmers. By 1960, there were about 13,000. By 1970, there were over 250,000, more than 600,000 by 1981, and, ultimately, there would be more than a million and a half programmers by 1993. See Appendix A.

previously been. Transistors replaced vacuum tubes, and transistors would soon be replaced by integrated circuits. Magnetic disk drives became available, with really large quantities of storage. Superimposed on these hardware inventions, and made possible by them, were the two inventions at the heart of system software: the operating system and the high-level programming language. System software permitted a programmer to write programs with minimal reference to the details of the machine; and in a language comparatively close to his own; and to collect, define, and use an appropriate special-purpose vocabulary. The net effect of all this was to control the complexity deriving from the computer itself.[1]

System software, and thereby the computer, became a locus of rational order, to which fragments of information continually attached themselves and thereby became orderly themselves. The accumulation was progressively sorted out, reorganized, consolidated, and cataloged to be used for other purposes. It was this accumulation, as much as the system software per se, which transformed the nature of programming. This control of complexity reduced the cost and difficulty of programming tremendously. Reducing the cost of programming made it feasible to program new applications, which were typically smaller, more diverse, more numerous, and more individually complex.

Easier methods of programming created an expansive mood. There were limitless numbers of things which could be done with

---

1. See Appendix B for an extended discussion of comparative complexity.

computers if they could be done cheaply enough. Computer programming assimilated new bodies of knowledge, and certain segments of computer programming came to define themselves as the art of assimilating new bodies of knowledge, as a kind of applied philosophy. As Herb Grosch, then DATAMATION's columnist, and eventually to be president of the Association for Computing Machinery, put it in 1960:

> ...we have to wade through the artifacts of a whole discipline like chemical reactor theory or airline reservations practice, or even find ourselves forced to codify and regularize where no set discipline exists... the provision of ...[programming] tools should occupy only a few of our professionals, not dominate the working thoughts of a majority... To pioneer a new application expands our horizon far more than to redefine the use of the semicolon in UGHTRAN, and should stimulate a far larger percentage of our best people.[1]

This might be called "systems analysis in the large," but there also existed a "minor systems analysis," which, as I shall develop later, consisted in formulating the practices of particular workplaces. Computer programming is fundamentally an incorporative discipline. This is in striking contrast with traditional engineering. Engineering disciplines had reached their limits of expansion when they had ceased to be within the realm of mutual comprehensibility; that is, when the demands of complexity had compelled different members to specialize in incompatible ways. Engineering disciplines were inherently

---

1. "Plus & Minus by Herb Grosch," DATAMATION, January/February 1960, p. 32

exclusionary, rejecting outlying bodies of knowledge.[1] Computer programming was remarkably free of this sort of exclusion because it did not have a complexity problem. The identity of computer programmers was tied up in looking outward for new things to program. These new tasks were generally forthcoming, and more and more programmers were hired to program them.

The extreme rate with which computer programming was expanding meant that employers found programmers where they could, and however they could. Often they raided each other's employees. Headhunting was a normal mode of recruitment. At the beginning level, no particular qualifications were required. There simply were not enough qualified people to go around. Employers found people where they could, and trained them to program. As late as 1986, company-financed programming training was still going on.[2]

Programming was open to the most marginalized groups of outsiders, such as the severely disabled, and prison inmates and ex-convicts. Intelligence and diligence were about the only requirements to become a programmer. Sometime in the fifties or sixties, Ida Rhodes of the National Bureau of Standards was giving instruction in computer programming to the deaf-and-dumb and blind. In 1972, IBM began a program, in cooperation with the state of Virginia, to train the severely disabled as computer

---

1. For the case of mechanical engineering, see: Monte A. Calvert, The Mechanical Engineer in America, 1830-1910: Professional Cultures in Conflict, The Johns Hopkins University Press, Baltimore, 1967, pp. 31, 39, 40, 127, 189-90, 215

2. See Janet Dight, "Grow Your Own Programmers," DATAMATION, July 1, 1986, p. 75

programmers. By 1986, a series of similar programs had produced more than 1900 programmers. About 1975, a similar program had been founded in England, organized by the British Computer Society. The disabled were intelligent, and they could be diligent when the circumstances justified, and so they became computer programmers. What applied to those immobilized by nature also applied to those immobilized by human ordinances. Even prison inmates could become computer programmers.[1]

By comparison, the alleged shortcomings of women were trivial. Whatever new problems women brought to the programming workplace were not intractable. Solutions were devised, often by using computer technology in new ways. Women had no significant difficulty in becoming programmers; they remained so in spite of motherhood; and in due course, they were promoted to management. Entrance into computer programming was the simplest part. There were simply not enough programmers. Formal educational programs were being set up, but that took time. As long as the number of programmers required was expanding exponentially, it was bound to outrun the supply.

The women who entered programming did so from a wide range of origins, and with a correspondingly wide range of qualifications.

---

1. Eric A. Weiss, "Biographies: Obituary of Ida Rhodes," Annals of the History of Computing, 1992, 14[2]:58-59; Robert J. Crutchfield, "News in Perspective-- Training: Back to Living," DATAMATION, September 15, 1986, p. 48, 52, 56; Nancy Foy, "International-- United Kingdom: Disabled Programmers Herald Self-Help Successes," DATAMATION, June 1977, pp. 168-CC, 168-FF; Jeff C. Larkin, "Letters [to the editor]: Inmate dp," (DATAMATION, February 1976, p. 7-8).

Some of them were highly educated in technical subjects; some were highly educated, but not in technical subjects; and some had minimal education.

Sometimes women got their chance by being secretaries in the right place at the right time. Such women had often had no previous chance to demonstrate ability at any technical or scientific work, or even to demonstrate the ability to deal in abstractions. As Mary Lynn McCaffery of Citibank put it:

> ...I became a secretary in a department that was involved in developing time sharing programs as a new product for correspondent banks. I found secretarial work frustrating and I kept pestering everyone for more to do. My boss suggested I try programming, so I borrowed a FORTRAN manual and started programming and found I really enjoyed it.[1]

Nobody really gave Grace Householder a chance to become a computer programmer-- she found the chance in a power vacuum and helped herself. Universities, hospitals, and kindred institutions are a bureaucratic anomaly. The nominal heads of an academic or medical institution have remarkably little disposition to actually run it. The institution is run by women of no great rank and no great authority: department secretaries, office managers, and suchlike. Grace Householder was an office supervisor in a small community hospital.

In 1966, Householder's hospital got a small computer (an NCR 500), and the vendor staked Householder to a minimal course of

---

1. "Women in Management: A Conversation," (DATAMATION, April, 1980, pp. 131-140), p. 134

training, lasting only a week, and presumably of the "this is a computer" variety.[1] According to the practice of an ordinary business, with efficient professional management, the programming of this computer might have been handed over to a consulting firm, to specially recruited expert programmers, or even to one of the ruling elite (physicians in this case). But this was not a business; it was a hospital. Householder simply went into action by herself. She managed to learn to program by herself, and having done so, went on to write programs to do various useful things around the hospital, and in due course to build up a data processing department.

Other women came into programming through the regular recruiting arrangements, especially those for college graduates. Christine Millen had studied Classics at the University of London. A Classics degree did not interfere with getting a programming job. As Millen puts it, "it wasn't necessary to have any special skills: the attitude was 'we'll take brains and train them.'"[2] Millen did not really set out to become a computer programmer. She interviewed with International Computers Limited; she was offered a job; she accepted it without any long-term intentions; and then one thing led to another.

Some women did have a technical background. Women engineers were rare, but there were a few of them. Donnamaie E. White had gotten a BSEE circa 1964. She had started out doing the

---

1. eds., "'Enthusiasm is Contagious,'" DATAMATION, September 1977, p. 30, 34

2. "Women in Management," loc. cit., pp. 133-34

electrical engineering she had been trained for, but had graduated into aerospace systems design. Later, she went to graduate school, specializing largely in software for electronics design, and even did a stint as a database project manager. By 1979, she considered herself "both a hardware and software person." She was then employed in developing components to improve the performance of old computers.[1]

Women mathematicians were more common. Nancy Jordan had gotten a Mathematics Degree from Wellesley, on the grounds that the employment prospects were better than Art History. As a Mathematics major, she was directed into scientific programming rather than business programming. Her first jobs were with IBM and later, Computer Sciences Corporation. Both jobs involved working on NASA contracts, starting with Gemini. Eventually, Jordan got bored with scientific programming, and put in for a job in Holland under contract to N. V. Phillips, working on an operating system. But she then decided she did not like living in Holland, so she came home again, and took a job in Computer Science Corporation's New York office, doing business programming. In due course, she moved on to Chase Manhattan, and went into management there.[2]

There were many possible ways to enter programming, open to many different kinds of people. Programming was expanding rapidly, as fast as it could expand, and it needed every last

---

1."People: Into Engineering Early On," DATAMATION, November 1977, p. 29-30

2. "Women in Management," loc. cit.,

person. In a couple of years or so, the novice had become a more-or-less experienced programmer, possessing what amounted to journeyman status. By this time, she often wanted children. Pregnancy and motherhood were not the kind of career disaster they might have been in other occupations. Women programmers found ways to work part-time, or otherwise to keep their hands in.

This had not been the case in other occupations which employed large numbers of women without becoming feminine ghettos. Journalism, for example, employed large numbers of women. By 1972, fully 41.4% of editors and reporters were women. By 1981, the figure was 50.1.[1] If gross numerical equivalence had been all there was to equality, women journalists would have been equal by the mid-seventies. But as late as 1988, when sexism as such was no longer acceptable, only thirteen percent of the directing editors were women. For a woman journalist to take time off in order to have children was commonly disastrous to her career. The top women were single or childless.[2]

Programming was different. Programmers were able to shape programming into a form of their own choosing. Not all programmers were women, let alone mothers, but almost all of them

---

1. Statistical Abstract of the United States, 103rd Edition, 1982-83, U. S. Department of Commerce, Table 651, "Employed Persons by Sex, Race, and Occupation: 1972 and 1981," pp. 388-89

2. Mary Ellen Schoonmaker, "The Baby Bind: Can Journalists Be Mothers" (Columbia Journalism Review, March/April 1988, pp. 33-39), p. 39.

valued their own autonomy. Most real-world constraints had been designed out. Still more constraints would be designed out in due course. The programming mother was commonly able to design a system of programming according to her needs, a system in terms of which she was not particularly unfit or handicapped.

A woman programmer who became pregnant was under no immediate obligation to quit work. Programming is no more physically strenuous or awkward than the traditionally approved knitting of infant garments, and it is considerably more lucrative. Likewise, as long as she was available to consult in an emergency,[1] the woman programmer could take a few weeks off to give birth. At this point, the programming mother had a range of choices. She could either continue to work full time (or even full time and then some); or she could find ways to work at home, treating programming as just another of the traditional domestic occupations of mothers.

Some women took the former approach. Christine Millen spent four weeks away from work, and returned with a promotion.[2] That was probably fairly atypical. Many mothers left work in whole or in part for much longer periods of time.

All else failing, women programmers with children could join together to organize their own software companies, in which their needs were axiomatically paramount. An example of such a company is the English firm, F International. F International was founded

---

1. To "consult" in this context generally means to explain where one put something, or what one did.

2. "Women in Management," loc. cit., p. 136

in 1962 by (Mrs.) Steve Shirley, to provide programming employment to women with children. By 1976, it employed 340 people, mostly married women with children dispersed over England and Scandinavia. They worked at home or in small dispersed offices, typically on a twenty-five hour week. F International specialized exclusively in custom software. The project managers were also dispersed, living near their major customers. At least at the managerial level, there was very little job rotation-- the emphasis was on cultivating ongoing relationships with customers.[1]

If F International was not formally a womens programming cooperative, then it was practically indistinguishable from one. By 1980, F International was up to 600 employees. By that time, there was a small American company run along the same lines, Heights Information Technology Service, started in 1978.[2]

Of course, these two firms were numerically insignificant, but that did not mean that they were insignificant in their influence. As Joan Wallach Scott has pointed out in her study of nineteenth century French glassworkers,[3] worker cooperatives cannot suspend the basic economic forces which act on a business.

---

1. "Calm View of Management," DATAMATION, August, 1976, p. 13-14.
   This is a profile of Suzette Harold, then-managing director of F International, and an exemplary long-term employee.

2. Edith Myers, "Home is Where the Work is", DATAMATION, February 1980, p. 77-79

3. The Glassworkers of Carmaux; French Craftsmen and Political Action in a Nineteenth-century City, Cambridge, Mass., Harvard University Press, 1974, pp. 167-87

Conversely, a functioning cooperative is unlikely to be doing anything an ordinary firm cannot do. Cooperatives, like other forms of labor diversion, are a less economically destructive form of strike, and their real significance is not in the number of employees, but in the terms of trade and work practices which they legitimize.[1] F International was a concrete demonstration that women could be usefully employed as programmers on their own terms. Because the company, or one like it, was ultimately available to discontented women programmers elsewhere, it defined the terms and conditions of employment that regular employers would be compelled to offer to women and mothers.[2]

Far more common than women's cooperatives were special arrangements within ordinary firms. Very small firms, with their lesser degree of bureaucratic inertia, were naturally more receptive to unconventional arrangements. For example, Integrated Software Systems Corp. was a small software firm in San Diego with five employees and one program. The company had been founded

---

1. For example see Herbert A. Applebaum, Royal Blue: The Culture of Construction Workers (Case Studies in Cultural Anthropology; ed. George and Louise Spindler; Holt, Rinehart, and Winston; New York; 1981) pp. 60-61, for a discussion of the importance of ownership of tools in permitting economic independence.

2. Parenthetical note: fairness, or equality, is about what terms or conditions people can justly be asked to accept, even though they don't want to accept them. Freedom, or autonomy, on the other hand, is about the idea that people can reject terms and conditions simply because they don't like them. If a given activity can be brought within the domain of freedom, fairness becomes irrelevant. Paying taxes is a matter of fairness. On the other hand, you cannot claim a sense of ill-usage because, when I make a copy of Linux for a friend, I do not send you a copy. You are at liberty to make your own copy because Linux is free.

about 1970, with the purpose of commercializing[1] some software which one of the principals had developed incidental to his Ph. D. dissertation in engineering. All five employees worked at home, using computer terminals, and accessing remote computers by night. There was no central office. By 1975, the team's chosen meeting place was the home of the one female employee and working mother, Sunni Harris. Although she was one of the two newest members of the team, Harris lived in a conveniently central location, so the team naturally met at her house. She had previously worked for Lockheed, one of Integrated Software Systems' customers.[2]

Small firms and large ones competed for programmers in the same labor market. Their conditions of employment were inevitably linked, especially if they did business with each other. Big companies made considerable allowance for the working mother. Mary Lynn McCaffery, who in 1980 had reached managerial rank at Citibank, had spent fully six years working part-time, circa 1970, while her children were young. At first, she had worked with a terminal, and gone into the office once a month; then later, she had gone to a three-days-a-week arrangement. When she was ready to resume full time work, it was only three to four months before the bank started promoting her into managerial

---

1. Commercializing software would mean taking some theoretically interesting but hopelessly defective software, and improving it to the point where it was no longer more trouble than it was worth.

2. E. M. (Edith Myers), "Working at Home and Liking It," DATAMATION, January, 1975, p.103-106

positions.[1]

When the woman programmer's children started school, she was still a programmer. At this point she might have several years of programming experience, and thus be ripe for promotion to manager. For women to make their way into management required them to convince their superiors that they were not going to quit suddenly in order to raise families. When a woman programmer expressed a desire to be considered for a vacant management position, an earnest discussion would follow, of the variety which Tracy Kidder calls "signing up."[2] The higher manager's attitude in such a discussion could be misconstrued as sexism, but it was probably more of devil's advocacy. The purpose of the discussion, as it would have been with a male subordinate, was to make sure that the candidate understood all the implications of the new job, in terms of responsibility, obligatory overtime, etc., and was prepared to accept them.

For actual or prospective mothers, discussion of child-care arrangements was necessarily included. Such discussion did not imply rejection of women pro women, or mothers pro mothers, but was merely an insistence that they think through the implications of combining a demanding professional career with motherhood. Christine Millen got her decisive promotion by declaring: "You don't have to worry about my home life. I'll worry about that--and I'm prepared to go abroad. I'm prepared to make the extra

---

1. "Women in Management," loc. cit., pp.132, 134

2. The Soul of a New Machine, Avon Books, New York, 1982, pp. 63-66 (orig. pub, Little, Brown and Company, 1981)

energetic effort on behalf of the company."[1]

Here again, the rate of expansion entered in. Many programmers were simply too inexperienced to be promotable. In the short term, the limiting factor for job opportunities was not so much a limited supply of jobs, but rather the limited rate at which employees could grow to fill those jobs. As long as job opportunities were expanding faster than trained ability to perform those jobs, competition between programmers remained at a fairly limited level (more recreation and learning device than Darwinian struggle for existence).

In the long run, the supply of jobs mattered too. Programming could expand numerically only if it also expanded conceptually, by continually broadening the range of things which could be programmed. If the range of what could be programmed was extended fast enough, that would provide room for the growth of the total number of programmers, for the individual growth of each programmer into a master craftsman or craftswoman surrounded by his or her journeymen and apprentices, and even for the growth of labor-saving devices such as advanced programming languages. But if this extension of the range of programming were to falter, then programming would slip into economic stagnation. The expansion inherent in the growth of each individual’s skills would soon carry the profession to the limits of its subsistence. Promotion would dry up, and great numbers of overqualified programmers would appear. People would begin to compete for each

---

1. "Women in Management," loc. cit., p. 136

other’s jobs, and this would soon become group competition, with racial, ethnic, and sexual groups all striving for their quotas of employment. All this baggage of economic decline could only be avoided by continually redefining programming outward.

The programming crisis noted by Aspray et. al. did not simply exist. It was socially constructed by decisions about what needed to be programmed. Probably the most numerically significant issue was "minor systems analysis," that is, the tailoring of code to particular workplaces. Minor systems analysis defined itself in opposition to package software. The working premise of package software was "one size fits all." The basis of minor systems analysis was a bias towards interpreting the world in nonmechanistic terms. A humanistic interpretation of work involved dealing with particular groups of humans. As Edward Yourdon eventually pointed out, these groups of humans would inevitably differ, and would therefore require different software, and their own programming staffs.[1]

At the same time, it was not inevitable that programming should adapt to institutional cultures. There was an oppositional tradition of what one might call business iconoclasm, an outgrowth of Taylorism. In the writings of Robert Townsend et.

---

1. For the programming crisis, see Peter Freeman and William Aspray, The Supply of Information Technology Workers in the United States, Computing Research Association, Washington, D.C., 1999.

Ed[ward] Yourdon, "A natural Productivity in Object Orientation," ch. 6 of: Software Engineering Productivity Handbook, ed. Jessica Keyes, McGraw-Hill, New York, 1993.

See Appendix C

al. and the practice of Ross Perot, one finds the idea that organizations not only develop their own ways of doing things, but also their own purposes and objectives, which may not be desirable. According to this reasoning, it is a good thing to periodically disrupt corporate cultures in order to refocus the organization around its stated purpose. Standard software becomes a valuable tool, and instead of programmers, the organizational reformer needs tough hatchet men who can force people to start over with the standard software, and the standard practices it embodies. The expansion of programming was based on the rejection of business iconoclasm.[1]

Women tended to play a considerable role in this outward redefinition of the goals of programming. They brought more than their share of mental breadth to programming. They were, par excellence, the collectors of new complexity from the outside world to replace that which had been subdued.

Women brought to computer programming qualities which filled the gap left by the male arch-technician. They were more likely

---

1. Robert Townsend, Up The Organization: How to Stop the Corporation from Stifling People and Strangling Profits (Fawcett Crest, Greenwich, Conn, 1971; orig. pub. Alfred A. Knopf, 1970). Revised edition published as: Further Up The Organization(Alfred A. Knopf, New York, 1984).

See also, Francis McInerney and Sean White, Beating Japan: How Hundreds of American Companies are Beating Japan Now-- and What Your Company Can Learn from their Strategies and Successes (Truman Talley Books/ Plume [Penguin], 1994), orig. pub. North River Ventures, Inc. 1993. A more recent development of the same ideas.

For Ross Perot, see Doron P. Levin, Irreconcilable Differences: Ross Perot versus General Motors, (Plume [Pengu],New York, 1990. orig. pub. Little, Brown, & Co., 1989), especially pp. 100-101, 136, 174-175, 206-207 (1990 edition).

to be broadly educated than men, and thus more predisposed to think broadly. Computer programming generated a new kind of technician. This new technician was not an engineer, but almost an anti-engineer. Unlike engineers, computer programmers were free to be versatile. They could look outward, using computer programming as a means to reinterpret the external world. The style of computer programming was lighter than that of engineering, with less premium on sheer bulldog tenacity, and more emphasis on receptiveness and imagination. Computer programming drew on all kinds of abilities traditionally cultivated by women. By making feminine characteristics into virtues instead of vices, programming evolved as a technical field uniquely receptive to women.

Women sometimes introduced humanistic concerns into the discourse of programming. For example, Angeline Pantanges, who in 1977 was an Associate Editor of DATAMATION, and international Editor by 1979, often wrote about the social issues of computing. We find her doing such things as systematically polling the leaders of the field to elicit statements that computers were meant to be about more than corporate data processing, warning against the excesses of European nationalism and English Thatcherism, and supporting Soviet dissidents.[1]

---

1. "Computer Wish List," (DATAMATION, January 1977, pp. 56-58) and "Social Concerns: Reminiscences and Predictions," (same issue, pp. 156-61) are about the real role of the computer. "No Charter for Human Rights" (ibid., September 1977, p. 244-A) and "Shcharansky's Agony" (ibid., October 1977, p. 150-54) deal with the case of the Russian dissident Anatoly Shcharansky. "Is The World Building Data Barriers?" (ibid., December 1977, p. 90-103)

As will be shown later, various women wrote about the issue of women in programming. However, whatever the long-range implications of a resident conscience like Angeline Pantanges, the most important literary role of women in the short run was to write about personal computers. Women did not play a leading role in developing the first personal computers[1], but they were the ones to bring personal computers back into the fold of computer programming.

In the mid to late seventies, advocacy of personal computers was hardly a way to win popularity contests. For one thing, personal computers seemed a rejection of the whole tradition of complexity-managing software which had grown up around computer programming. Personal computer software remained primitive for some time to come. The programming languages which could run on such small computers had almost none of the features required to control and manage complexity. For the first few years, personal computers were open to technical scorn. As late as 1980, it was still possible to speak of personal computers in quite unflattering terms. Dorothy A. Walsh remarked that "Personal

---
...Continued...

is about the threat of European data nationalism. "They Could If They Would" (ibid., June 1979, p. 194-A) is about the privatizing excesses of the Thatcher government in Britain.

1. Though there were exceptions like Lore Harp and Carol Ely of Vector Graphic, who started their own home business (specifically, in the bathroom of Ms. Harp's home) making a memory board for small computers which Ms. Harp's husband, an engineer at Hughes, had designed for them.
See: Adam Osborne, Running Wild: The Next Industrial Revolution, Osborne/McGraw-Hill, Berkeley, California, 1979, pp. 33-34

computers are fun. They are joining hi-fi, do-it-yourself kits, and movies as favorite pastimes. They are useful. But they are not computers in the classical sense." She referred to the personal computer programmer as a "self-made 'programmer,'" and a doer of mathematics homework.[1] The gibes of people like Walsh were annoyed responses to the promoters of personal computers. DATAMATION's pages had been infiltrated by people who were engaged in businesses involving personal computers. Far more of them were women than was the case in personal computer manufacturing proper.

Word processors were a kind of incipient computer. The better free-standing word processors had all the hardware components of a personal computer, and were personal computers for all purposes and intents. Amy Wohl was not the only person to write about word processing at the time, but she was somewhat unconventional in choosing to cover free-standing word processors. While other commentators stressed eventual integration of word processing into a central or networked system of some kind, Wohl promoted the free-standing type, a here-and-now de-facto personal computer.[2]

---

1. Dorothy A. Walsh, "Forum: A Modern Alladin's Lamp," DATAMATION, March 1980, pp. 272-74

2. See her "What's Happening in Word Processing" (DATAMATION, April 1977, pp. 65-74) and her later "Word Processing 1979: A Market Evaluation" (DATAMATION, May 25, 1979, pp. 112-14).
By way of contrast, see J. Christopher Burns, "The Evolution of Office Information Systems," (DATAMATION, April 1977, pp. 60-64) and Robert B. White, A Prototype for the Automated Office," (same issue, pp. 83-90), both of which exhibit the big-system mentality.

Still more forceful in her argument was Portia Issacson, co-owner of a computer store, who had a regular column, "Personal Computing," in DATAMATION, starting in October 1977. Her first column began:

> Personal computing is computing so low in cost that it can be used abundantly, even wasted! Personal computing is a revolution in the availability of computing resources. No longer will information processing be available only to large corporations and government; we'll soon have computing power at out fingertips in out homes and offices... The coming widespread availability of the personal computer ranks with other great technology-based revolutions-- the printing press, the assembly line, and the automobile.[1]

Issacson became more outspoken over time. By 1979, she was talking about personal computer owners starting to program their computers, and thus being able to launch an autonomous body of software of their own;[2] and calling for the equivalent of a mainframe computer on every desk.[3]

Amy Wohl and Portia Issacson were leading figures in positioning DATAMATION to go into personal computers. By so

---

...Continued...

The main locus of social construction in computer systems lay in this sort of choice, rather than in the act of invention. Most inventions never reach prototype stage; most prototypes never reach niche market; and most niche products never become widely used. By opting for the freestanding word processor, Wohl was in effect choosing the personal secretary (19th century executive apprentice or 20th century office wife models) as a social norm, and rejecting the ideology of office as paper factory.

1. DATAMATION, October 1977, p. 210

2. "1979-- The Year of the Home Computer," DATAMATION, January 1979, pp. 217-18

3. "Personal Computing: 1984's Information Appliances," DATAMATION, February, 1979, pp. 215-218

doing, they helped to prevent the mainline programming community from defining itself into a corner as successive groups of engineers had done over the years. Personal computers would have happened whether DATAMATION and its readers liked them or not, but if data processing centers had consistently stuck to a hostile attitude and refused to cooperate with personal computer users, there could have been considerably more disruption than there was.

Women had promoted the idea of social responsibility, and they had promoted the dangerously unpopular idea of the personal computer. It was no great leap to promoting themselves, that is, to promoting the idea that women as a group could be programmers, and do well at programming. The public image of the woman programmer gradually caught up with the facts, often running ten years or more behind reality. During this period, women programmers were making immense gains, not in percentages (which had already been achieved), but in absolute numbers. This advance was not noticeably hampered by the fact that, in the early years, the public sphere of the computer press, and especially the advertisements, still treated women in a nonserious fashion.

In the early sixties, sexism was still blatant and outspoken, though even then it tended to attach itself to women who were not programmers. In 1959, a group of RAND Corporation employees, including E. A. Feigenbaum and F. J. Gruenberger, had written, in a letter to the chairman of the Western Joint Computer Conference: "It is ridiculous to have to sweat out a long line, then get your card to fill out, then find that the little girl

has never before met the problem of making change..." As the preliminary advertisement for the 1960 Western Joint Computer Convention put it: "Delegates Attending the 1960 Western Joint Computer Conference may cheerfully neglect their wives for business-at-hand, knowing that a well-rounded program of social events has been organized expressly for the benefit of the ladies." A cartoon of "The Well Dressed Programmer" published in 1965 showed a recognizable grid (ie. gunsight-style) etched on one of the (male) programmer's eyeglass lenses. The caption explained that this was for "sizing up secretaries."[1]

Women programmers were collectively invisible. The programming of the early sixties did not overflow with public roles, and no doubt each woman programmer, within her own sphere, tended to be treated as a special case, with no great awareness that there were enormous numbers of special cases out there. The shortage of programmers ensured them a fair chance to show their ability, and once shown, they could live on their earned individual reputations. However, the material basis for even this kind of ineffectual sexism was being steadily undermined.

A transition point was reached in an unusual little piece

---

1. Feigenbaum et. al. to Robert M. Bennett., 15 July 1959, published in: "And Still More on Computer Conferences: Letters to and from WLCC's Chairman" DATAMATION, January/February, 1960, pp. 43-45, see p. 45; Announcement of the WJCC: DATAMATION, March/April, 1960, p.48; After the fact report, DATAMATION, May/June, 1960, p.23; "The Well Dressed Programmer," DATAMATION, Jan 1965, p. 58

about a small plastic model computer, the DIGI-COMP I.[1] The piece was written by a fourteen or fifteen year-old schoolgirl, Karen Ann Schneider. Miss Schneider was by no means the first female DATAMATION writer. Established professionals had gained the recognition of publication in the main trade journal. But she was in all probability the first female DATAMATION writer who could not possibly fit into the category of honorary male. She alternated between the conventional role of stage moppet, familiar to watchers of old situation comedies; and something new, a person engrossed with a computer, an incipient hacker. The tone of the short blurb provided by the editors tends towards the tone customarily used to describe children playing at being adults. The woman-as-little-girl image was more or less spontaneously disintegrating before one's eyes as soon as a computer was attached to it. Women programmers were coming to recognize themselves, not as extraordinary women who had nothing in common with ordinary women, but rather as representative women, blazing a trail which other women could follow if they wished. In the March 1, 1971 issue of DATAMATION, a letter from Pamela A. Henline called for the recruitment of women programmers, the creation of part time positions for mothers, and so forth.[2] These demands had in practice already been met, one woman programmer at a time. But no one had talked about it. Woman programmers could have little sense of their own numbers. Now, as

---

1. Karen Ann Schneider, "DIGI-COMP User Report," DATAMATION, January 1965, p. 55

2. p. 13

everyone pooled their knowledge, they came to realize that there were a lot of women programmers.

In July 1980, Gloria Steinem, who was then probably the most eminent of American feminists, wrote a short article in Ms. magazine, "Rx Fantasies: For Temporary Relief of Pain Due to Injustice."[1] In it, she constructed a series of scenarios whereby women might somehow seize concentrations of male power (the Mideast oilfields, the Pentagon, the New York Times, etc.), and begin using this power for feminist ends. Most of these scenarios ran on the premise that men owned the bastions of power, and could simply refuse to discuss the matter; hence the scenarios generally involved tricking the men into surrendering this power. For example, it was an obvious fantasy that the pope would be willing to stake his whole prestige on a single public debate with a feminist. However, one of the scenarios featured a group of cleaning women reading up on computer technology, and then conducting a computer heist to appropriate the wealth of the large banks. For Steinem computers were apparently part of the

---

1. Ms. magazine, July 1980, p. 99, reprinted in Gloria Steinem, Outrageous Acts and Everyday Rebellions, Signet (New American Library), New York, 1986, orig. pub. 1983, pp. 341-345.

Steinem, who had founded Ms. eight years earlier, had begun her journalistic and feminist career by working as a Playboy Bunny and then writing about it. This provided the informing paradigm of her life's work, the remark that "...all women are Bunnies (Outrageous Acts, p.78)." Certainly, it is remarkable that the editor of a well-known magazine could think of the New York Time's circulation as simply given, rather than as something to be won away. Surely, her own experience must have taught her that no one need subscribe to or read a publication if they choose not to.

Steinem was certainly more representative of the broad currents of American feminism than such essentially hard-boiled, tough-minded "business feminists" as Jane Trahey or Jo Foxworth.

male structure of domination. The fantasy was not so much that bank computers could be raided-- computer crime was very much in the news by then.[1] The fantasy was that women could do such a thing; could actually do such a male technological thing as programming a computer.

At that time, seemingly unbeknownst to Gloria Steinem, there were, at a rough estimate, a quarter of a million women gainfully employed as computer programmers in the United States.[2]

There was a long distance between encouragement of women programmers and concurrence with the full rhetoric of the women's movement. When Anne Petrokubi reproached the editors[3] for describing a successful woman executive as comely, and for supposedly denigrating feminist militancy, the editors replied, coolly, that the offending item was both written and edited by women. Likewise, Bernard Galler, who taught Computer Science at the University of Michigan, reported on the explosive anger of his female students when a recruiter gave the impression that they counted as affirmative action hires.[4] This was a far cry from the "right to affirmative action to make up for past discrimination" doctrine put forward by some feminists.

Feminism tended to get absorbed in the broader ideology of

---

1. For example, see Thomas Whiteside, Computer Capers: Tales of Electronic Thievery, Embezzlement, and Fraud, 1978, republished as a Mentor paperback in 1979.

2. See Appendix A

3. Letters, DATAMATION, February 1976, p. 8

4. Letters, DATAMATION, June, 1977, p. 223

computer programming. When Gonnie Siegel attacked unconscious sexism,[1] her arguments wound up being redeployed in support of the whole community of programmers, female and male. Ms. Siegel was a management consultant, specializing in feminist issues, and past president of her local National Organization of Women chapter. It is not clear whether she was a programmer, but she was probably not a very highly qualified one. Her analysis was not really formulated around programmers, or engineers, for that matter. It was more based around the general mentality of the businessman. She attacked discrimination against women not as wrong, but as illogical. Discriminatory managers, who prided themselves on their rationality, were in fact in the grips of another cluster of emotions, which can be approximately glossed as machismo. One element of this machismo merged into pride in rationality, especially as expressed in ruthlessness. Only by overcoming their sexism and machismo could managers become truly rational.

Much of this was not directly relevant to technical management, and especially not to computer programming. As we have seen, in computer programming, provided it kept expanding, there simply was not the demographic basis for any meaningful sort of discrimination, on the basis of sex or any other irrelevant ground. However, calling into question the traditional behavior of businessmen created common ground with other people who might be at odds with it. Thus, one of the letters in reply

---

1. Gonnie Siegel, "The Forum: The Best Man for the Job May Be a Woman," DATAMATION, June, 1976, pp. 196-200

to Siegel's article, from Peter Martin,[1], was about the liberating effects for a man of not having to pretend to "business macho," along with the "ulcers, heart attacks, and other dysfunctions" it would produce. Feminism was turning into general anti-authoritarianism. Even the seemingly promising agenda of pay discrimination was elusive. As Christine Millen remarked in 1980:

> Maybe I was paid a thousand or so less than a man at times, but there was never any substantial differential as far as I could determine... No large corporation can afford to discriminate... [and] in large corporations, you *do* find [them] out.[2]

To this, Molly Nemhouser replied:

> Contrarily, I've worked for smaller companies; information like that has been more secret, more protected. When I found out I wasn't making an equal salary, I asked for more money. And I got it, no problems, no questions asked.[3]

Women programmers might exemplify the highest goals of feminism, but that did not necessarily make them receptive to the more extreme feminist ideas, with their emphasis on perpetual oppression. Rather, women programmers were coming to recognize themselves as they had in fact been since the beginning, neither as special cases, nor as oppressed victims of sexism, but

---

1. Letters, DATAMATION, September, 1976, p. 7
2. "Women in Management," loc. cit., p. 139
3. ibid.

instead, as highly autonomous professionals.

Computer advertising lagged behind editorial content for a while, but eventually, the image of women in advertisements caught up. At first, the advertisements treated women in a simple-minded way. For example, there were the Computape advertisements, circa 1962, in which a girl named Penelope flirts with a reel of tape.[1] Blatantly sexist advertisements persisted up to about 1970. Deborah Sojka of DATAMATION later collected an assortment of these advertisements.[2] For the late sixties, there are: a picture of a pregnant woman, with the caption that her mechanical replacement won't get pregnant and be incapacitated by morning sickness; a United Airlines ad with a pun on the "stewardess" or "coffee, tea, or me" joke; and the inevitable girl in a bikini. The worst offenders refused reprint permission, obviously hoping that would be the end of it. But Ms. Sojka was not going to let them get away that easily. Instead, she printed their citations and brief (fair use) quotations of their advertising copy. For example one firm had said, in 1970, "We taught our data entry system to speak a new language: Dumb Blonde."

But that was one of the last instances of sexism. In the early seventies, advertisements began to catch up with the editorial content of the mid-sixties, and the social actualities of the mid-fifties. The Association for Computing Machinery was one of

---

1. DATAMATION, February 1962, p 62; August 1962, p. 74

2. "The Old Clothes of Advertising", DATAMATION, September 1982, p. 137-53

the first organizations to change over. By 1971, it had used, for a membership recruitment advertisement, a profile of a recent new member, a woman systems programmer.[1] But that was something of an exception, remote from the normal pattern of Id-driven selling.

The more characteristic advertisement of the early-to-mid-seventies was an analog of the editorial treatment of Karen Ann Schneider. That is, advertisements depicted woman in the process of being transformed from little girl to adult by the action of computers.

A Computer Terminal Corporation advertisement suggests: "scare hell out of your secretary. Get her a computer. About 45 minutes after the handsome thing is on her desk, she'll be an expert." There were a series of time-sequence photographs of the secretary going through a phase of terror, having hysterics, and so forth; but eventually winding up with a delighted smile on her face.[2]

A Digital Equipment Corporation advertisement offered: "...A big computer system that's small enough for anyone," and shows a little girl taking a computer out of the boxes, putting it together, and eventually sitting down in front of it.[3]

Then the advertisements began to examine the relationship

---

1. ACM advertisement, DATAMATION, April 15, 1971, p. 86
2. DATAMATION, March 15, 1971, p.43
3. DATAMATION, July, 1977, pp.38-39

between computers and power. A Prime Computer advertisement[1] depicted four women doing various interesting things using computers. One of them was a hard-core ('real time') systems programmer; another was doing an actuarial simulation; a third was doing a literature search; and a fourth was writing a manual. Interestingly, while the last two are wearing conventionally feminine clothing and posed in conventionally demure attitudes, the hard-core programmer is wearing slacks, sleeveless tunic, sensible shoes, no make-up, and is standing in an assertive posture which may or may not be a fencer's stance. The statement was quite clear: there was a place for women who wanted to remain conventionally feminine, and quite a good place by most standards, but there were also additional opportunities for amazons.

As the seventies came to an end, commercial advertisements rapidly trended towards the tone the ACM had adopted at the beginning of the decade: matter-of-fact, with equality taken for granted. A Bell and Howell advertisement showed a woman troubleshooting a printer with the company's new recording oscilloscope.[2] An Informatics Inc. advertisement offers a testimonial from a woman systems analyst.[3]

In one of the most striking advertisements, Intertel, a maker of modems and computer networking devices, turned the very idea

---

1. DATAMATION, August 1976, p. 38
2. DATAMATION, January 1976, p. 135
3. DATAMATION, January 1979, p. 19

of sex appeal on its head to make a nasty joke at the expense of a rival. They showed a picture of the kind of sequin-clad blonde who might have starred in a daytime game show, and commented:

> The 1960's. A face. A smile. And buy, buy, buy... You could take some starstruck kid from Kankakee. Dress her up like a fashion model... And every time she smiled, cash registers all over America would start to chime... It wasn't just toothpaste, either... Take the folks at Milgo, for example. They sold modems that way. A lot of modems.
>
> But all of a sudden, the sixties were gone. And the 1970's rolled in like a wave. People discovered they had networks. Not just modems and terminals (They also discovered that women weren't sex objects, but that's another story)... While Intertel was busy building network control systems, Milgo kept right on building modems... And they waited for the orders to roll in. While they were waiting, Intertel introduced them to the rough and tumble world of marketing in the 1970's.[1]

In short, sexism is not an exercise in robbing women of their rights, but merely the mark of an incompetent fool. At that point, sexism in computer advertising was unequivocally dead. It was not politically incorrect, but factually incorrect. An insistence on the false premise of sexism would entangle one in a series of other unrealities, as well as the disasters which go with departure from reality. By the year 1980, when Gloria Steinem was writing her fantasy about cleaning women, women programmers had attained a series of degrees of freedom and equality. They had become numerous. They had become economically independent, and if pay scales for women programmers were not equal yet, equality was fast approaching. Women programmers had

---

1. DATAMATION, March 1977, pp. 10-11

found autonomy in their work, either as highly skilled professional employees, or by running their own companies. They had attained both individual and collective respect within the profession, at both the conscious and subconscious levels.

This victory had not been attained by affirmative action or quotas, nor by political agitation, nor by efforts to ban the expression of derogatory sentiments. It had been gained by harnessing the potential of the computer. Each development in the computer’s technology had implied a social corollary, and on the whole, the corollaries had run in the direction of human freedom.

Now, in 1980, another computer revolution was impending, that which Portia Issacson had proclaimed. The next phase would be that of the personal computer, and it, too, would have its corollaries.

Appendix A: Statistical Table

year	1960	1970	1972	1976	1977	1981	1983	1987	19
source	1973	1973	1982 H		1978	1982	1994	1989	19
Both sexes, in 000's *********									
Total Workers									
Engineers	864	1210	1111	1150	1267	1537	1572	1731	17
Civil	157	173	156	160	171	190	211		2
Industrial			171	187	214	237	210		2
Math. and Comp. Sci.							463	685	10
Comp. Sys. Anal., Sci							276	447	7
Ops. & Sys. Res. & Anal.				124			142		2
other							45		
Computer Specialists	13	258	276	363	371	627			
Comp. Sys. Anal.				122					
Computer Programmers				223			443	527	5
Total Eminent Programmers(*)				246			418	640	10
Total Programmers(*)	13	258	276	487	371	627	861	1167	15
Computer Equipment Op							605	914	6
Computer Op							597	911	5
Off. Machine Op.	239	423	679	714	759	966			
Computer & Periph Op			199	295	302	564			
Key Puncher	122	253	284	250	280	248			

year	1960	1970	1972	1976	1977	1981	1983	1987	19
percent women **********									
Total Workers	32.8%	38.0%	38.0%	39.6%	40.5%	42.8%	43.7%	44.8%	45
Engineers	--	--	0.8%	1.1%	2.7%	4.4%	5.8%	6.9%	8
Civil	--	--	0.6%	1.3%	1.2%	1.6%	4.0%		9
Industrial			2.4%	2.7%	7.0%	11.4%	11.0%		16
Math. and Comp. Sci.							29.6%	34.1%	32
Comp. Sys. Anal., Sci							27.8%	32.1%	29
Ops. & Sys. Res. & Anal.				15.3%			31.3%		39
other							35.3%		36
Computer Specialists	31%	20%	16.8%	21.2%	23.2%	38.5%			
Comp. Sys. Anal.				14.8%					
Computer Programmers				25.6%			32.5%	36.6%	31
Total Eminant Programmers(*)				15.1%			29.0%	34.0%	32
Total Programmers(*)	31%	20%	16.8%	19.3%	23.2%	38.5%	30.8%	35.2%	31

Computer Equipment Op							63.9%	66.0%	61
Computer Op							63.7%	66.0%	61
Off. Machine Op.	V.A.	V.A.	71.4%	69.5%	73.8%	73.6%			
Computer & Periph Op			37.8%	44.5%	55.5%	63.8%			
Key Puncher	V.A.	V.A.	89.8%	92.8%	93.2%	93.5%			
year	1960	1970	1972	1976	1977	1981	1983	1987	19
number of women, in 000's ********									
Total Workers									
Engineers	--	--	8.9	12.6	34.2	67.6	91.2	119.4	147
Civil	--	--	0.9	2.1	2.1	3.0	8.4		20
Industrial			4.1	5.0	15.0	27.0	23.1		33
Math. and Comp. Sci.							137.0	233.6	340
Comp. Sys. Anal., Sci							76.7	143.5	229
Ops. & Sys. Res. & Anal.				19.0			44.4		93
other							16		
Computer Specialists	4	51	46.4		86.1	241.4			
Comp. Sys. Anal.				18.1					
Computer Programmers				57.1			144.0	192.9	182
Total Eminant Programmers(*)				37.0			121.1	217	323
Total Programmers(*)	4	51	46.4	94.1	86.1	241.4	265.1	410.0	505
Computer Equipment Op							386.6	603.2	373
Computer Op							380.3	601.3	369
Off. Machine Op.	239	423	484.8	496.2	560.1	711.0			
Computer & Periph Op			75.2	131.3	167.5	359.8			
Key Puncher	122	253	255.0	232.0	261.0	231.9			
year	1960	1970	1972	1976	1977	1981	1983	1987	19

Sources:
---- 1960, 1970: Statistical Abstract of the United States, 94th Annual Edition, 1973, U. S. Department of Commerce, Table 375, "Experienced Civilian Labor Force...," pp. 235-239
For 1960 and 1970, occupations are subdivided as male or female. An occupation itemized for one sex may not be so for the other. In the table of percentages of female workers, 'V. A.' (virtually all) means that the occupation was not listed for men. '--' means that it was not listed for women. Percentages, and total number of workers are based on the cited numbers for male and female workers, and incorporate undetermined round-off error.

---- 1972, 1981: Statistical Abstract, 103rd Edition, 1982-83,

Table 651, "Employed Persons...," pp. 388-89

---- 1977: Statistical Abstract, 99th Edition, 1978, Table 681, "Employed Persons...," pp. 419-421

---- 1983, 1993: Statistical Abstract, 114th Edition, 1994, Table 637, pp. 407-09

---- 1987: Statistical Abstract, 109th Edition, 1989, Table 642, pp. 388-389

---- 1976:
Louise Kapp Howe, Pink Collar Workers

Blank spaces indicate categories not listed for that year.

For 1972 onwards, the Statistical Abstract gives numbers of workers in thousands, and fraction female in tenths of a percent. I have computed from these the number of female workers.

I have calculated the total number of programmers, total, female, and percentages female; and where there is sufficient data, I have also calculated totals of 'Eminant Programmers,' those superior to the ordinary programmer.

The figures for the numbers of Eminant Programmers in 1987 are based on the assumption that the number of persons included in 'mathematical and computer scientists,' but not classified more specifically in 1983 and 1993-- presumably pure mathematicians, about 45,000 of them-- did not change appreciably.

In fifty years, from 1940 to 1990, the total number of computer programmers grew from almost nothing to more than a million, despite massive increases in programmer productivity, and at any given time, about a quarter to third of those programmers were women. This was usually about five times the proportion of women in engineering, and only somewhat below the proportion of women in the labor force.

Parenthetically, computer programmers as a whole are not to be confused with computer scientists, a much smaller and more academically oriented fraction of the profession. Definitions are notoriously treacherous, but a working definition of a computer scientist would perhaps be a programmer who writes programs involving pointer variables and recursion, that is, making use of the contents of the typical sophomore computer science course in algorithms and data structures. Most programmers have been qualified or apprentice systems analysts, something quite different.

For 1970, the Statistical Abstract provides average salary data for programmers: male: $11,193, female: $7,763, or 69.4% of male pay (all pay figures are in 1969 dollars). However, given that for this date, no information whatever is offered concerning relative age, seniority, credentials, skill, etc., these figures must be regarded with the utmost caution.

However, taking the weekly earning in that year for all

workers over 25 years (men: $148, women: $88, or 59.5%), we find that the pay differential in programming may have been slightly less inequitable than was customary in the larger society. In any case, the salary figure for women programmers worked out to $149 per week, substantially comparable to the average man’s pay. A female programmer’s annual income was comparable to that of a construction craftsman ($7,660); higher than any other female occupation listed; and approximately double the female average of $3,649. The female programmer was emphatically making a ’family wage,’ even if she was not making an ’engineer salary.’

Appendix B: Complexity in Computer Programming and Conventional Engineering

Programming is comparatively well partitioned, and its complexity is more under control than that of conventional engineering. By contrast with programming, engineering is unpartitioned. The famous Rube Goldberg cartoons are well partitioned machines: their comedy lies precisely in the extravagant disproportion between costs and results. Thinking about, or designing, a nonpartitioned mechanical system in a partitioned way means progressive departure from reality. Complexity is unavoidable in engineering. Even those branches of engineering, such as electrical engineering, which are intellectually closer to computer programming are still embedded in engineering. Paul Ceruzzi, in "Electronics Technology and Computer Science, 1940-1975: A Coevolution" (Annals of the History of Computing, 1989, 10[4]:257-275), lays stress on the extension of the idea of complexity from computer science into electrical engineering, and the progressive reduction of electrical engineering to a science of information processing.

However, one must point out that electrical engineering students are still required to meet the general requirements of the engineering school, in which the department of electrical engineering is embedded. That is, under the rubric of 'engineering fundamentals,' they must cover the equivalent of an undergraduate major in physics, with great chunks of effort put into subjects like chemistry, physical chemistry, thermodynamics, hydraulics, etc., which are very remote from the new information science oriented conception of electrical engineering.

Further, an engineer who lacked the relevant bench or field skills (eg. machine shop work), typically learned on the job, tended to be ineffectual because he could not directly put his ideas into practice. Compared to a multi-year machinist's apprenticeship, even learning machine language was a comparatively minor difficulty. Eventually, after 1980, of course, many engineers ceased to be engineers per se, and became programmers of engineering programs such as CAD/CAM systems. This was especially the case for engineering scientists, that is, engineers specializing in the more mathematical aspects of engineering.

Finally, one must remember that complexity has a social dimension. To take one example, an airplane wing is simultaneously an airfoil; a loadbearing structure similar to a bridge; a fuel tank; and an equipment locker. Under the traditional precomputer modes of engineering, designing a wing was of course far more work than one person could do. Since most of the components of the wing are multi-purpose, it is effectively impossible to design the various "aspects" in isolation. Any change in one aspect dictates changes in other aspects. Aircraft design is notorious for complex synergistic bugs of a type which are rarely seen in computer software.

In practice, aircraft wings were likely to be designed under the "tyrant" system, meaning that the chief designer gave out assignments to subordinates working in one big room, and then wandered around, intervening in their work without warning. Nevil

Shute, who was of course a notable aircraft industry manager before he became a novelist, took the view that a successful aircraft designer was necessarily an aggressive bully. See his No Highway (1948, William Morrow; paperback: Ballantine Books, New York, various editions, inter alia 1969, ch. 2 ,pp. 46-48). Shute's disclaimers notwithstanding, the character E. P. Prendergast is a thinly disguised Barnes Wallis, Shute's old chief. In such a regime, anyone not prepared to work substantially the same hours as the chief was likely to be regarded as a nuisance for not being there when wanted, and therefore to be driven out. Since the chief was, almost by definition, a workaholic, the established norm was likely to be the 80-hour week. In effect, such a chief engineer treated his subordinates as a human-powered CAD/CAM system.

The novel thing about programming was the extent to which an equally driven manager could simply break out subprograms, specify the parameters, assign them, and not worry about these subprograms very much until the subordinates had completed them. Similarly, the manager could exercise his workaholism at home by reading and red-penciling printouts as late into the night as he wanted.

Appendix C:

Ed[ward] Yourdon, in "A natural Productivity in Object Orientation," (ch. 6 of: Software Engineering Productivity Handbook, ed. Jessica Keyes, McGraw-Hill, New York, 1993) treats package software as a comparative novelty. However, contra Yourdon, package software, in the largest sense of the word, is rather older than is generally realized. Service bureaus were a form of package software, which happened to come with its own computer and operating staff.

John L. Roy ("The Changing Role of the Service Bureau," Datamation, March 1970) and Richard H. Hill ("That Big Bucket in the Sky," loc. cit.) deal with the blurred role of the service bureau as a vendor or franchiser of this emerging package software. While this code could of course be modified, doing so was not a major priority, and the emphasis was on keeping deviations from standard practice down to the minimum. Leonard J. Palmer, in his Computer Center Finance and Operation, (1970, MMKS inc., San Francisco) develops elaborate cost estimates for a service bureau, reckoning "systems and programming" at 20% of total revenue, compared to 40% for keypunching and 30% for the computer and its operation (p. 0-2). In this regime, programming naturally shaded off into cost estimation.

Of course, package software can become a programming language in its own right, but that simply begs the question of standardization or customization.